Open Genius:

Email Metrics and Consumer Privacy

1

Consumer and government scrutiny of digital privacy and privacy rights is rising. Apple has addressed privacy in email by launching ‘Mail Privacy Protection’, caching all email images automatically, masking device information and generating false open signals for all iOS 15 MPP recipients.

In presenting consumers with an either-or choice — privacy or email open-rate data — Apple is destroying the leading indicator of campaign engagement. Email marketers have depended on this signal for decades.

Open Genius delivers a balanced solution, restoring open measures and open rates and ensuring consumer privacy.

Open metrics are available live for campaigns from any email platform, without integration or data exports. Data sovereignty laws are adhered to through the application of edge computing and encryption. No PII is involved.

Open Genius is a use-based subscription, available at https://campaigngenius.io/OpenGenius.

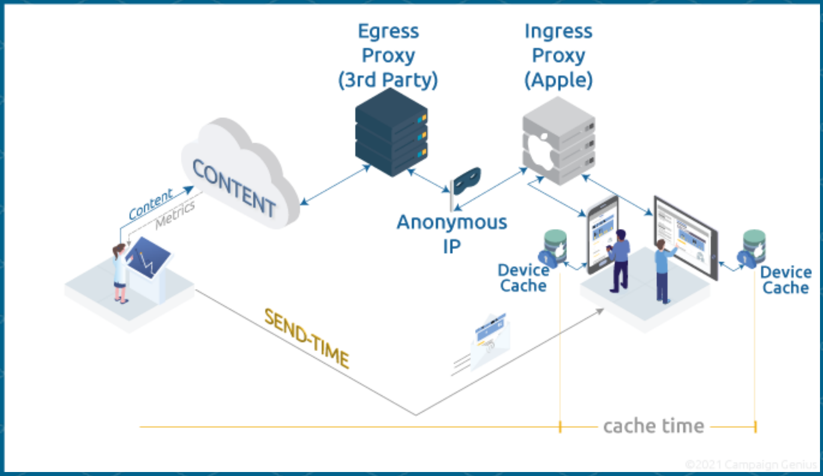
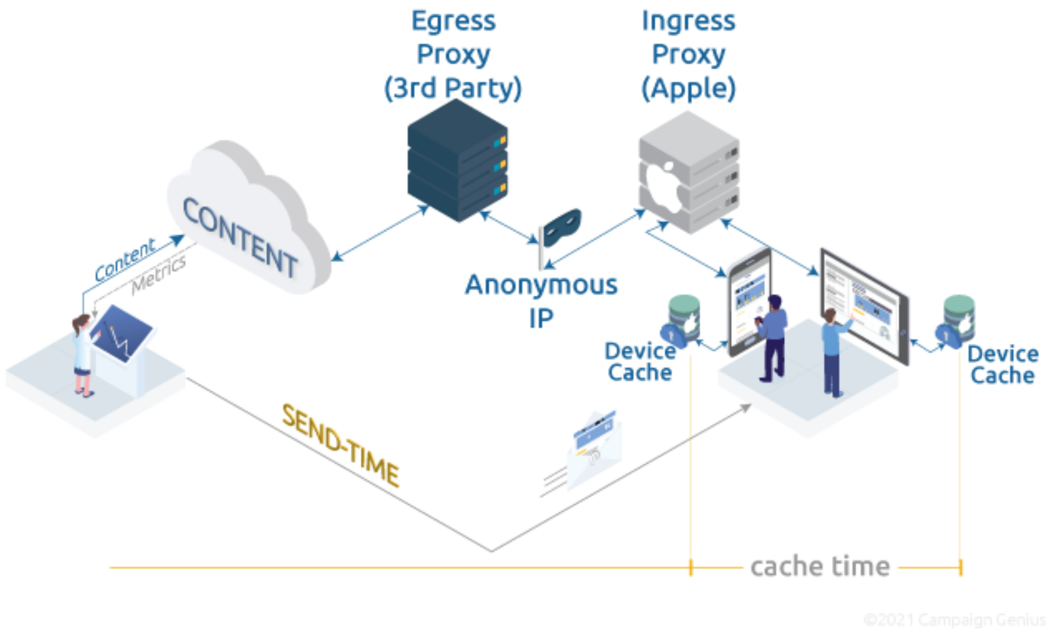


Table of Contents

1 INTRODUCTION

Apple's Email Privacy Protection, announced in June 2021, aims to "stop senders from using invisible pixels to collect information about the user." Apple will be implementing a complex 2-tier system to retrieve images/pixels and obscure user IP address. Once fetched, content is cached on-device.



As Apple devices account for nearly 50% of email traffic (Litmus), this change would have widespread impact on key marketing measures — most notably, open rates — as well as email features like real-time content and personalization.

The email marketing industry has concluded — wrongly, in my opinion — that open rates are dead, aka "Pixelgeddon."

Open Genius is a service from Campaign Genius that delivers projections of opens and open rates, including user actions hidden in MPP traffic. This white paper describes the data and math involved. (This material is covered by one or more patents pending.)

2 The Technology and Data Playing Field

Open rates are an historical accident, the result of a “technology credit.” When HTML was included as a standard message-body type over 20 years ago, HTML image tags (<img>) using the Web HTTP protocol for source fetch were included. This inclusion created a mechanism — probably unintentionally — to observe user email actions through the robust data provided by the HTTP protocol.

The image tag has been used to deliver visible images, of course. But the nearly ubiquitous “tracking pixel” — a small, invisible-to-the-user image — became predominant use of that tag, at least by count. (Real-time content, which manipulates image content dynamically, runs a far, faint third.)

As awareness has grown of incursions into privacy in the digital age, email and ‘tracking pixels’ started getting public focus. The ‘Hey’ email service (from Basecamp), launched in 2020, pitching itself as a rethink of email, promising ‘no spy pixel tracking’ (despite the presence of Plausible Analytics tracking scripts on Hey.com).

Apple, self-branded champion of privacy and operator of a multi-billion-dollar advertising division, announced “Mail Privacy Protection” in June 2021. Because Apple iOS devices are so widely used, and because the native iOS email client is so commonly used on those devices, MPP threatens to render email opens and open rates useless.

The heart of the issue lies in Apple’s approach to email privacy. Rather than just relaying image requests through a proxy server (as Gmail has done for nearly a decade), Apple has designed a complex, multi-tier caching system. This caching system scans email message bodies and fetches image content without user intervention, eliminating useful time-of-open data or correlation with user action. The fetch request is relayed through a 2-tier infrastructure, masking both the HTTP User-Agent header (which identifies browsers, devices and other characteristics), and the requesting IP address.

As nearly 50% of email messages are read on iOS devices, this approach presents a flood of “false open” signals. Simplistic open/send calculations are becoming increasingly meaningless. The open rate — the broadest and fastest indicator of campaign engagement — is at risk.

Open Genius is designed to help marketers with this challenge while fulfilling the consumer-privacy intent of MPP.

3 Open Genius Design

3.1 Data Protection & Edge Encryption

Open Genius tackles the privacy/rate challenge, paradoxically, by adding data. Open Genius-enabled images (pixels, images, or real-time image content) require 3 data elements to work:

- Unique identifier for a collection of sent emails (e.g. per-campaign)
- Unique identifier for each recipient (e.g. customer ID, email address, or GUID)
- Identifier value for the Open Genius customer.

The 2nd item — customer ID — would seem to contradict the notion of consumer privacy, so let’s address that at the top.

Open Genius does not actually store the unique recipient ID, or use it to resolve opens and open rates. Instead, by design, recipient IDs are “hashed” with the SHA256 algorithm. SHA256, designed by the US National Security Agency and published in 2001, is a reliable, one-way encryption function. The 256 bits resulting from a SHA256 hash are essentially indecipherable; “it would take 10 * 3.92 * 10^56 minutes to crack a SHA256 hash using all of the mining power of the entire bitcoin network.”[*] Each recipient ID, in other words, would take some 7^51 years to crack.

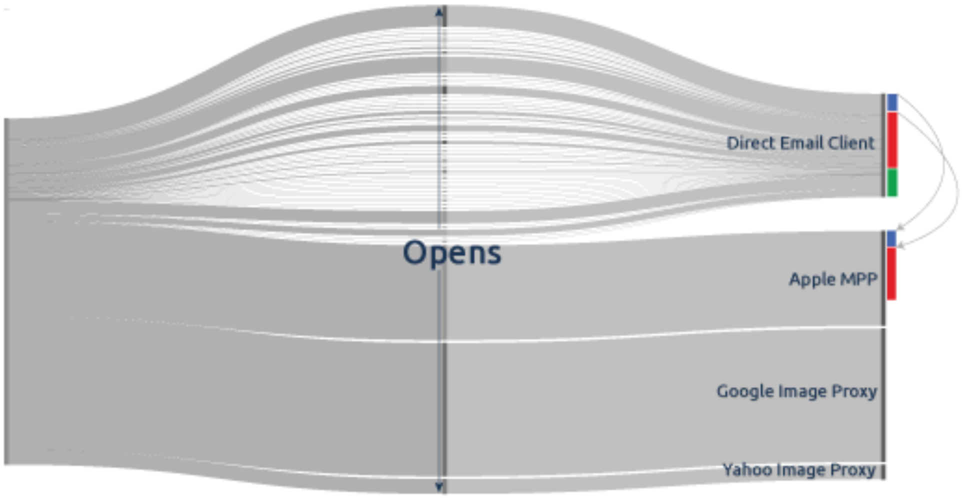
To further protect privacy, Open Genius handles ID encryption “at the edge.” Using (ironically) the CDN provider employed by MPP, Open Genius requests are handled in over 250 data centers worldwide. Relevant request information, including recipient IDs and IP addresses, are hashed “in-country” at the nearest point of presence. Hash values are then encrypted and forwarded for storage; request data such as recipient ID are discarded.

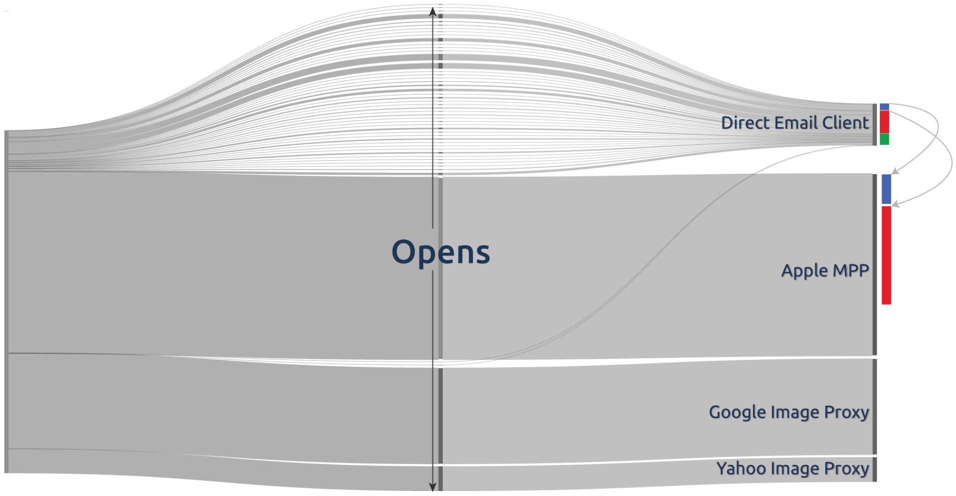
[*] https://bitcoin.stackexchange.com/questions/41829/wont-asic-miners-eventually-break-sha-256-

3.2 How Are The Data Used To Project Opens?

The essential argument behind Open Genius is that, given a statistically-significant sample, “people are people” — that open behavior for a given campaign/subject line that can be measured directly excluding MPP ‘false opens’ can be projected on the MPP false opens.

The key word here is “project”; opens and rates are statistics-based, not recipient based.





Put simply, Open Genius generates a by-device model of email engagement for a campaign from the non-MPP recipients, and applies that model to the proportion of recipients in the “black box” of MPP false opens.

That’s still a mouthful, so let’s break it down a bit.

(1) Identify MPP Requests

Through the iOS 15 beta, and up to Oct 2021, Apple's MPP provides several data signals that identify it as the source of a particular request. One such signal is the User-Agent string included in the HTTP request header. MPP provides 'Mozilla/5.0' as a User-Agent value. It's meaningless, but it's distinct and it's consistent — at least thus far.

Apple also publishes the IP addresses of the network-egress points of MPP requests[†] — some 325,664 CIDR networks as of Oct 2021. The majority are IPv6, with effectively an infinite number of egress addresses after CIDR expansion.

Between these two signals, MPP requests can be identified relatively reliably.

Campaign Genius developers have identified a 3rd signal based on Internet network architecture. Validation tests are underway as the company evaluates patent prior art.

(2) Identify (Anonymous) Recipients

In opting to fetch every image (sooner or later), Apple's MPP removes the user moment-of-open from the data stream. That makes sense, as a privacy protection move. From a data-set perspective, though, it provides a signal that says "this recipient receives email on an iOS 15 device."

Because the hash algorithm is consistent, a recipient ID is anonymous but reliable. If I receive an email on iOS 15, and also open it on Exchange, the same hash will be calculated for both opens.

These two sets of data — recipient ID hashes from requests, and MPP requests — can be combined to isolate MPP recipients within the set of open requests. For example, my Exchange open request can be excluded from the set of requests, because it has the same user ID hash as my iOS 15 MPP request.

As a result, the data from recipients who do not use iOS15/MPP can also be isolated and characterized separately. We can, in effect, study people with iOS 15 devices, and those without, as separate groups within a given campaign.

[†] https://developer.apple.com/support/prepare-your-network-for-icloud-private-relay

(3) Derive User-Agent Patterns

The User-Agent HTTP header is a very loose 'standard', with no authority or enforcement body. A developer can devise and send an arbitrary user-agent string at will; presumably someone named Will Norris did so with the "willnorris/imageproxy" user-agent values in our database. There are reportedly tens of thousands of user-agent strings in use.

While it is a loose standard, however, there are services available to resolve user-agent strings into useful data. Open Genius incorporates this. User Agent string 'Mozilla/5.0' from MPP aside, the user-agents from other requests can be used to group and characterize them.   How many desktop requests? Smartphone? Tablet? Other devices?

In other words, user-agent strings provide the data to sort out device-related behavior for sub-groups within a campaign. What % of users opened an email more than once? How many on a smartphone and a desktop? A desktop and a tablet? And so on.

Other sub-groups can be extracted as well (assuming a large enough sample). How did Apple users not yet on iOS 15/MPP react to this campaign? Android users? How did people who opened a campaign message on Day 1 do so?

(4) Handle Proxies

Thanks primarily to the popularity of GMail, email image requests through proxy servers are relatively common. Proxied requests do not provide "what kind of device" data, so they're not useful in characterizing groups as just mentioned. But because they can be clearly identified, they can be excluded from calculations about sub-group and device behavior.

That doesn't mean that proxied requests aren't useful in other calculations, though. They factor in calculations about proportion and time, in particular. As long as Google Image Proxy continues to fetch images based on user open, Gmail will provide stats that, along some dimensions, help balance out the opacity of MPP.

That narrative describes the key steps — the actual data, queries and calculations are considerably more complicated.

(5) Summary

Reduced to bullets, Open Genius flow is roughly this:

- Separate MPP and non-MPP recipients.
- Isolate non-proxied requests within non-MPP requests.
- Calculate the proportion of devices (desktop, smartphone, tablet, 'other'), and the statistical margin of error of this set.
- Calculate the relative proportion of MPP "false opens" and true opens.
- Project the smartphone + tablet opens (proportionately) in the MPP group.
- Aggregate non-MPP recipient requests, MPP recipient direct requests, and projected MPP requests together.
- Divide by sends for projected open rate.

The queries and calculations involved can be done for a given campaign, at a given (live) point in time. That mitigates a number of non-obvious risks.

One such risk: MPP behavior. As of Oct 2021, MPP is not auto-fetching images instantly. A calculation of opens and rates based on historical information, or user records, would be thrown off by this system behavior. Because Open Genius calculations are live, however, a "non-open" doesn't affect results. If I have read an email on Exchange, and my iPhone has not yet MPP-fetched the pixel in that email, I will show up as a non-MPP open. If my iPhone fetches the pixel overnight, that open will show up on the "MPP direct open" side of the ledger tomorrow, and the MPP false-open will factor in the projection tomorrow.

Conversely, if the system behavior of MPP changes — if the fetch schedule becomes more aggressive, for example — calculations will self-adjust. The presumptions about MPP that matter are (a) opens don't indicate user behavior and (b) MPP opens can be identified in some way.

It's also worth noting that the data and structures aren't MPP-specific. Industry rumors say that another inbox provider — possibly Yahoo — might follow Apple's lead. The methodology can be 'split' to map the model to other providers. The caveat, of course, is that the sample size used for modeling behavior would get smaller.

4 Observations on Privacy

It’s worth noting that all recipient-related identifiers, and all IP addresses, are hashed; all recipients are anonymous, not just MPP users.   Request location (based on IP) is recorded. In the wrong hands, this could be ‘spun’ as an invasion of privacy. Our view is that aggregated locations, with no PII, are aggregate measures — “how many where”, not “who where.”

Because Open Genius does not dictate the identification token for recipients, it is also inherently not designed to aggregate conclusions about recipients across campaigns. Individual customers, such as email marketers, are in control of the recipient identifier; Open Genius just aggregates hashes.