

## ENGINEERING

November 9th, 2016 | Updated February 23rd, 2023

# How we built link tracking to be reliable, fast, and secure



[Andrew Theken](#)

SHARE

We recently released a new enhancement to email sending called “Link Tracking” (see [our announcement blog post](#)). With this feature, you are now able to get more insights into how your recipients are interacting with the emails you send them, including their preferred browsers, where they’re located, and whether they prefer Text or HTML.

Most of the “add-on” features that we’ve provided in the past are unobtrusive; recipients usually won’t notice when Postmark is providing the feature. We want your use of Postmark to be completely invisible to your recipients. The only thing they should notice is how fast the email got to their inbox. Your recipients should feel confident that the the messages originate from you, and as far as you’re concerned, Postmark should be an “Implementation Detail.”

Just as an example, when you enable Open Tracking, a reference to a transparent image is added to your HTML emails. This image is loaded from one of our servers at <http://pstmrk.it>. If there is latency in loading this image, or if open tracking would become unavailable, your recipients shouldn’t notice since the tracking image doesn’t play a significant role in the content of the email being sent to your recipients.

help

# A new set of challenges

Link Tracking is a little bit different. This new use case is an example where a Postmark feature will take on greater prominence in the content of your emails, so this introduced new considerations in how we delivered this feature to you and your recipients.

Unlike more passive features, Link Tracking comes in to play when your recipients actively click a link, so this needs to be fast, reliable, and as transparent as possible. We also know that with cloud email providers, your recipients may search for and click on links that you send months or years from now, so it was important to us that tracked links you send today work for the foreseeable future. Finally, due to the nature of transactional emails, many tend to have user-specific information, such as password reset emails which usually include a reset link to the application. Our Link Tracking system needs to be as secure as possible, to prevent eavesdropping, and other information leaks.

Today, we wanted to share some of the technical details about how we built Link Tracking, and some of the decisions we made to provide a reliable, fast, and secure system.

## How link tracking systems work

First, let's talk about the general premise behind link tracking systems. Generally speaking, most "link tracking" and "short url" systems work based on a fairly simple flow:

1. Convert the original url to a url that points at a tracking server. This can take several forms, but a simple example might be: "<https://example.com>" converted to "<https://tracking-service.com/e...>"
2. When a user clicks on "<https://tracking-service.com/e...>", an initial HTTP request is sent to the "[tracking-service.com](https://tracking-service.com)" web server by a browser.
3. The web server typically records basic information about the request:
  1. User's IP
  2. The "User-Agent" header of the request (what browser they're using)
  3. The requested link
  4. The timestamp of the request.

4. The web server issues a 301 or 302 http status, which directs the user's browser to the original url (<https://example.com>).

There are variations on the above flow, but the general principle of routing a request through a tracking server is a widely-used and reliable method of tracking click activity. This is the basic technique used for Link Tracking in Postmark. The diagram below provides an overview of the process.

A quick overview of how link tracking works.

## Making link tracking reliable

As discussed previously, one of our design goals was that tracked links are very highly available, even under circumstances where other parts of Postmark may be experiencing delays or other issues ([which we work really hard to avoid, and to be transparent about](#)). To accomplish this, we designed *Link Routers* which process the click requests and redirect users to the original URLs in your emails.

The *Link Routers* are completely decoupled from the rest of our infrastructure. The first critical step to accomplishing this is that each tracked link includes enough information to be completely standalone. That means when we encode links, we add some meta-data which allows us to verify the validity of the link.

Here's an example of a link that has been converted:

`https://example.com` becomes

`https://click.pstmrk.it/2s/example.com/AQ/-----w/bTkMOXzMBY`

There is some extra gobbledygook in the encoded link, but we made every effort to minimize this and still make it possible to get a sense of what the original link looks like. We'll get into some additional details below to cover why we think this format makes sense.

When a tracked link is clicked, an *Link Router* receives and decodes the request, captures the LinkID, User-Agent, and Requesting IP to a local database. At the same time, the original request is immediately re-routed to the original URL, even before the click information is committed to the local database. Our thinking was that getting the recipient to the original url is more important than committing the data to the database, which can complete asynchronously. If there was some sort of issue that prevented storing a click to the database, your recipients are still arriving at the original urls as fast as possible.

In a background task on each *Link Router*, logged request information is synchronized to our central data centers periodically. If the sync API is not available for some time, the logged data remains on the *Link Router* until the API becomes available. This ensures that your recipients still arrive at the original urls, even if there is a delay in synchronization to our backend. This capability was unintentionally tested due to the DNS issues that a large part of the Internet experienced earlier this month. As soon as the DNS issues were resolved, the synchronization caught up, but during the DNS disruption, tracked links were still handled properly, without a delay.

We deploy *Link Routers* in pairs in three different regions throughout the world, and balance between the instances using HAProxy, so that we can do "[Blue-Green deployments](#)" to update the *Link Routers* without service interruptions.

# Making link tracking fast

As stated above, making link tracking fast and transparent was another important design goal. We used two techniques to achieve this goal.

First, as shown above, links are completely standalone. Our tracking system does not have to consult or synchronize any database in order to operate because the links themselves contain enough information on their own. This is in contrast to link shortening services that capture the original link in their own database, and then when a request comes in, need to lookup the LinkID to redirect you to the original url. Not relying on a database has several major benefits:

1. **Faster sending:** Links can be issued without requiring database transactions, so adding tracked links to outgoing messages adds only a few milliseconds to sending time for each message. Since TTI is critical, this is a huge benefit.
2. **Faster redirects:** Since there is no need to lookup a url, everything can happen in process, making Link Tracking “CPU-bound”, not “I/O-bound.”
3. **Strong Consistency:** Because there is no database to synchronize throughout the world, tracked links work as soon as they are issued.
4. **Simplicity:** No database to synchronize is one less thing to break!

The other technique we used to make things fast was to geo-distributed our *Link Routers*. We’ve been using [Packet](#) for some of our infrastructure and have been very happy with their service. Packet provides bare-metal machines that can be provisioned in a few minutes. The big benefit of bare-metal is that “noisy neighbors” can’t impact our application as easily, which makes our performance more predictable. Packet has data centers in three regions around the world: *Amsterdam*, *Newark, NJ*, and *San Jose, CA*. We currently have deployed *Link Routers* in each of these regions. Through DNS, we route incoming requests to the closest data center, ensuring that the latency for each request is minimal.

Due to the simplicity of the *Link Routers*, each instance is capable of handling about 1,900 requests per second, and we can also easily add capacity as Link Tracking usage grows. (For the curious, we used a tool called “[siege](#)” to load test these servers.)

# Making link tracking secure

Finally, and most importantly, we designed Link Tracking to be secure. Here's what we did:

First, we use full-time https; You're probably using TLS to send via SMTP or our API, which ensures no eavesdropping can happen between your application and our system. Where possible, you're probably also using https:// for the links you send.

In your original urls, using *https* prevents eavesdropping when a recipient follows a link. If our tracking system replaced these links with http equivalents, we would be opening a security hole where anyone running basic tools could "sniff" out these urls on their local network (in coffeeshops or public hotspots, for example).

Instead, we decided that all links being tracked by Postmark would use https. If the original url uses http, we will redirect using plain http, but the communication between a recipient's browser and our *Link Routers* will be encrypted. This decision has two main downsides:

1. It's not currently possible for you to set a "vanity domain" (such as "tracking.example.com") domain for tracked links. Certificates for HTTPS connections are associated with specific domains, so this means that using https does not allow us to support arbitrary CNAMEs for tracking domains.
2. Tracked Links will not work in IE6 on Windows XP. This is due to the fact that certain HTTPS encryption protocols and cipher suites that are known to be insecure are the only ones that IE6 supports. We limited the suites we are supporting to achieve an ["A" rating from Qualys SSL Labs](#).

We feel that both of these are reasonable trade-offs to ensure the security and privacy of links you send to your recipients.

Another important aspect to the security of our system is that the *Link Routers* do not store link information. What this means is that the *Link Routers* themselves do not contain a database of tracked links that could be stolen. Each link that is routed through the *Link Routers* is decoded on-the-fly, and only a "LinkID" is recorded for each tracking event. It is impossible to derive the original link from the LinkID. Additionally, and unlike typical web server setups, we explicitly avoid logging request information, which would be another potential vector for stolen data. Again the data isn't there to be stolen.

Finally, because we don't use a database of links on the *Link Routers*, we need a way to validate that each request to our *Link Routers* is valid. We do this by formulating an ["HMAC"](#) of the original URL and including it with the tracked link.

In the example link above, *bTkMOXzMBY* is the HMAC of the url. When we decode the original url, we verify that it matches the HMAC we issued before redirecting the link. Changing a single character in the tracking link will invalidate the link, preventing it from redirecting. Using this technique, we can ensure that we only redirect links that were issued by our sending system, and that our *Link Routers* cannot be used by third parties to mask the origin of requests.

There is no such thing as “perfect security,” but we feel that the measures above provide reasonably good assurances that your tracked links remain private.

## A long road

Link Tracking is a feature that has been requested for a long time. Like the rest of Postmark, we want you to be able to consider this feature a part of your infrastructure, enabling it, and not worrying about whether it works. For Link Tracking, we wanted to take time to really consider the best design to achieve *Reliability*, *Speed*, and *Security*. We hope that this intentional effort is evident as you enable Link Tracking for your most important communications.

---

SHARE

---

## Andrew Theken

Dominator of board games, avid reader, and public library advocate.

[@atheken](#)

# Email best practices and industry news. Delivered monthly.

Our **monthly** newsletter is packed full of email tips, product announcements, and interviews with industry experts.

Email\*

Please verify your request\*



I'm not a robot

reCAPTCHA  
[Privacy](#) - [Terms](#)

Subscribe

## More in Engineering

---

[How to use Postmark as your SMTP email provider with Mastodon](#)

[ActiveCampaign Hackathon Highlight: Postmark and AI](#)

[Increasing Postmark's capacity: A parable of pipes](#)

[Goodbye Colocated Data Centers: How We Moved Postmark to AWS](#)

[Managing your Postmark templates with Github using Travis CI or CircleCI](#)

### ◀ [Putting webhooks to work](#)

By Patrick Graham

### [Feature announcement: enhanced bounce webhook](#) ▶

By Rian van der Merwe



## Product

Pricing  
Customers  
Reviews  
Dedicated IPs  
iOS App  
Referral Partner Program

[Latest Updates](#)

## Features

Email API  
SMTP Service  
Message Streams  
Transactional Email  
Email Delivery  
Templates  
Inbound Email  
Analytics & Retention  
Integrations  
Webhooks  
Security  
Email Experts  
Rebound

## Postmark For

Agencies  
Startups  
Enterprise  
Bootstrapped Startups  
Side Projects  
Developers

## Postmark vs.

SendGrid  
SparkPost  
Mailgun  
Amazon SES  
Mandrill

## Resources

Blog  
API Documentation  
Getting Started  
Email Guides  
Email Comic  
Videos  
Podcast  
DMARC Digests  
Webinars  
Labs  
Migration Guides  
Newsletter  
Glossary

## Help

Support Center

Contact Support

Talk to Sales

Service Status

## Visit ActiveCampaign for:


Marketing Automation

CRM & Sales Automation

Landing Pages

SMS Automation

---

Made with  at

[Privacy Policy](#)

[Cookie Policy](#)

[Terms of Service](#)

[EU Data Protection](#)

© ActiveCampaign, LLC, 2025.