Characterizing Pixel Tracking through the Lens of Disposable Email Services

Hang Hu, Peng Peng, Gang Wang
Department of Computer Science, Virginia Tech
{*hanghu, pengp17, gangwang*}@*vt.edu*

***Abstract*—Disposable email services provide temporary email addresses, which allows people to register online accounts without exposing their real email addresses. In this paper, we perform the first measurement study on disposable email services with two main goals. First, we aim to understand what disposable email services are used for, and what risks (if any) are involved in the common use cases. Second, we use the disposable email services as a public gateway to collect a large-scale email dataset for measuring email tracking. Over three months, we collected a dataset from 7 popular disposable email services which contain 2.3 million emails sent by 210K domains. We show that online accounts registered through disposable email addresses can be easily hijacked, leading to potential information leakage and financial loss. By empirically analyzing email tracking, we find that third-party tracking is highly prevalent, especially in the emails sent by popular services. We observe that trackers are using various methods to hide their tracking behavior such as falsely claiming the size of tracking images or hiding real trackers behind redirections. A few top trackers stand out in the tracking ecosystem but are not yet dominating the market.**

I. Introduction

An Email address is one of the most important components of personally identifiable information (PII) on the Internet. Today's online services typically require an email for account registration and password recovery. Unfortunately, email addresses are often unprotected. For example, email addresses used to register online social networks might be collected by malicious third-parties [45], thus exposing users to spam and spear phishing attacks [40]. Massive data breaches, especially those at *sensitive* services (*e.g.*, Ashley Madison [22]), can expose user footprints online, leading to real-world scandals. In addition, email addresses are often leaked together with passwords [51], [56], allowing malicious parties to link user identities across different services and compromise user accounts via targeted password guessing [57].

As a result, *disposable email services* have become a popular alternative which allows users to use online services without giving away their real email addresses. From disposable email services, a user can obtain a temporary email address without registration. After a short period of time, the emails will be disposed by the service providers. Users can use this disposable email address for certain tasks (*e.g.*, registering an account on a dating website) without linking their online footprints to their real email addresses (*e.g.*, work or personal email). In this way, potential attacks (*e.g.*, spam, phishing, privacy leakage) will be drawn to the disposable addresses instead of the users' real email accounts. Disposable email services are highly popular. For example, Guerrilla Mail, one of the earliest services, has processed 8 billion emails in the past decade [3].

While disposable email services allow users to hide their real identities, the email communication itself is not necessarily private. More specifically, most disposable email services maintain a *public inbox*, allowing any user to access any disposable email addresses at any time [6], [5]. Essentially disposable email services are acting as a public email gateway to receive emails. The "public" nature not only raises interesting questions about the security of the disposable email service itself, but also presents a rare opportunity to empirically collect email data and study *email tracking*, a problem that is not well-understood.

In this paper, we have two goals. First, we want to understand what disposable email services are used for in practice, and whether there are potential security or privacy risks involved with using a disposable email address. Second, we use disposable email services as a public "honeypot" to collect emails sent by various online services and analyze *email tracking* in the wild. Unlike the extensively-studied web tracking [29], [34], [43], [48], [9], [10], [18], email tracking is not well-understood primarily due to a lack of large-scale email datasets. The largest study so far [17] has analyzed emails from 902 "Shopping" and "News" websites. In this paper, we aim to significantly increase the measurement scale and uncover new tracking techniques.

Understanding Disposable Email Services. In this paper, we collect data from 7 popular disposable email services from October 16, 2017 to January 16, 2018 over three months. By monitoring 56,589 temporary email addresses under popular usernames, we collect in total 2,332,544 incoming email messages sent from 210,373 online services and organizations. We are well aware of the sensitivity of email data. In addition to working with IRB, we also take active steps to ensure research ethics such as detecting and removing PII from the email content and removing personal emails. Our analysis reveals key findings about the usage of disposable email services.

First, there is often a delay to dispose of the incoming emails. Certain services would hold the emails for as long as 30 days, in spite of the claimed 25 minutes expiration time. Second, we find that users are using disposable email addresses to register accounts in a variety of online services. While the vast majority of emails are spam and notifications, we did find a large number of emails (89,329) that are used for account registration, sending authentication code, and even password reset. Third, accounts registered via disposable emails are easily hijackable. We find risky usage of disposable email addresses such as registering sensitive accounts at financial services (*e.g.*, PayPal), purchasing bitcoins, receiving scanned documents, and applying for healthcare programs.

Measuring Email Tracking. Email tracking involves embedding a small image (*i.e.*, tracking pixel) into the email body to tell a remote server when and where the email is opened by which user. When the email is opened, the email client fetches the pixel and this notifies the trackers. To measure email tracking in the wild, we build a new tool to detect both first-party tracking (where the email sender and the tracker are the same) and third-party tracking (where the email sender and the tracker are different) from the collected email dataset.

We have three key observations. First, email tracking is highly prevalent, especially with *popular* online services. Out of the 2.3 million emails, 24.6% of them contain at least one tracking link. In terms of sender domains, there are 2,052 sender domains (out of 210K domains in our dataset) ranked within the Alexa top 10K. About 50% of these high-ranked domains perform tracking in their emails. Second, we find that stealthy tracking techniques are universally preferred, either by falsely claiming the size of tracking images in HTML or hiding the real trackers through redirection. Popular online services are significantly more likely to use "stealthy" tracking techniques. Third, although a small number of trackers stand out in the tracking ecosystem, these trackers are not yet dominating the market. The top 10 email trackers are used by 31.8% of the online domains, generating 12% of the tracking emails. This is different from web tracking where one dominating tracker (*i.e.*, Google) can track user visits of 80% of the online services [31].

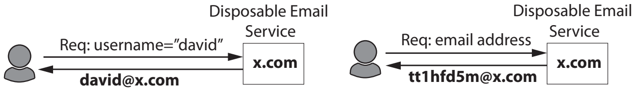
Contributions. Our work makes three key contributions.

- *First*, we perform the first measurement study on disposable email services by collecting a large-scale dataset (2.3 million emails) from 7 popular services over 3 months.
- *Second*, our analysis provides new insights into the common use cases of disposable email services and uncovers the potential risks of certain types of usage.
- *Third*, we use the large-scale email dataset to empirically measure email tracking in the wild. We show the stealthy tracking methods used by third-party trackers collect data on user identifiers and user actions.

II. Background

A. Disposable Email Services

Disposable email services are online web services where users can obtain a *temporary* email address to receive (or send) emails. After a short usage, the email address and its messages will be disposed by the service provider. Disposable email services allow users to register an online account without giving away their *real email addresses*. This helps to disconnect the user's online activities from her real identity, and avoid attracting spam emails to the real email accounts.



(a) User-specified Address. (b) Randomly-assigned Address.

Fig. 1: Two types of disposable email addresses.

There are two types of disposable email services, based on how temporal email addresses are assigned (Figure 1).

- **User-specified Addresses (UA).** Most services allow users to specify the username they want to use. For example, a user can obtain a temporary address "david@x.com" by specifying a username "david". The user-specified address is more memorable for users.
- **Randomly-assigned Addresses (RA).** Some services create temporal email addresses for users by randomly generating usernames. For example, a user may be assigned to a random address that looks like "tt1hfd5m@x.com". Users may refresh the web page to receive a different random address each time.

While disposable email services allow users to temporarily use an email address, this email address and the received messages are not necessarily "private". More specifically, most disposable email services are considered to be public email gateways, which means any users can see other users' temporary inbox. For example, if a user A is using david@x.com at this moment, then another user B may also access the inbox of david@x.com at the same time. Very few disposable email services have implemented the sandbox mechanisms to isolate each temporary inbox. The only service we find that maintains a private inbox is inboxbear.com, which distinguishes each inbox based on the browser cookie. Therefore, many disposable email services have made it clear on their websites (or Terms of Services) that the email inbox is *public* and users should not expect privacy [6], [5].

B. Email Tracking

Email tracking is a method that allows the sender to know whether an email is opened by the receiver. A common method is to embed a small image (*e.g.*, a 1×1 pixel) in the message body. When the receiver reads the email, the image will be automatically loaded by sending an HTTP or HTTPS request to a remote server. The remote server can be either the original email sender or a third-party service. In this way, the remote server will know when (based on timestamp) and where (based on IP) the email is read by which person (based on email address) using what device (based on "User-Agent").

Email tracking is part of the broader category of web tracking. Web tracking, typically through third-party cookies and browser fingerprints, has been extensively studied [15], [29], [34], [43], [12], [46], [48], [28], [19], [9], [10], [18], [38]. However, very few studies have systematically examined email tracking because real-world email datasets are rarely available to researchers. The largest measurement study so far [17] collected data by signing up for "Shopping" and "News" websites to receive their emails. The resulting dataset contains 902 email senders. The limited number and category of online services severely limit researchers' ability to draw generalizable conclusions.

We believe that the disposable email services provide a unique opportunity to study email tracking at a much larger scale and uncover new tracking techniques in the wild. First, disposable email services are *public*, which allows us to collect emails sent to disposable email addresses. Second, users of disposable email services have broadly exposed the email addresses to the Internet (by registering various online accounts), which helps to attract emails from a wide range of online services (and spammers). The resulting data, even though still has biases, is likely to be much more diversified.

III. Data Collection

To understand how disposable email services are used, we collect emails that are *sent to* disposable addresses. First, we describe our data collection process. We then present a preliminary analysis of the dataset. Finally, we discuss the active steps we take to ensure research ethics.

A. Data Crawling Methodology

Since disposable email addresses are public gateways, our method is to set up a list of disposable email addresses and monitor the incoming emails. In this paper, we primarily focus on *user-specified* addresses for data collection efficiency. We select a list of "popular" usernames which increases our chance to receive incoming emails. In order to increase our chance of receiving incoming emails, we select a list of "high frequency" usernames. Disposable email addresses under such usernames are often used by multiple users at the same time. In comparison, monitoring *randomly-assigned* (RA) addresses did not return many incoming emails. For example, in a pilot test, we monitored 5 RA email services (eyepaste.com, getnada.com, mailto.space, mytemp.email, and tempmailaddress.com) for 5 days. We only succeeded in collecting data from getnada.com and all inboxes in other RA services were empty. In total, we scanned 194,054 RA addresses, and collected 1,431 messages from 1,430 inboxes (a hit rate of 0.74%). The reason for the low hit rate is that randomly-assigned addresses come from a much larger address space than user-specified ones. Accordingly, in this paper, we focus on *user-specified* addresses for data collection.

Selecting Disposable Email Services. We spent a few days searching online for "disposable email" and "temporary email" to find popular services. This process mimics how normal users would discover disposable email services. By examining the top 100 entries of the searching results, we find 31 disposable email services (19 UA and 12 RA services[1]). UA services are typically more popular than RA services. For example, the top 5 sites have 4 UA services and 1 RA service. As discussed above, we focus the on services that offer user-specified addresses (UA), and select the top 7 disposable email services as shown in Table II. These services are very popular. For example, *guerrillamail.com* self-reported that they have processed nearly 8 billion emails in the past decade. *mailnesia.com* self-reported that they received 146k emails per day. While most of these services only provide the functionality of receiving emails, a few (*e.g.*, guerrillamail.com) also provide the functionality of sending emails. In this work, we only focus on the incoming emails received by the disposable email addresses (to analyze email tracking).

Selecting Popular Usernames. We construct a list of popular usernames to set up disposable email addresses. To do so, we analyze 10 large leaked databases (that contain email addresses) from LinkedIn, Myspace, Zoosk, Last.fm, Mate1.com, Neopets.com, Twitter, 000webhost.com, Gmail, Xsplit. These databases are publicly available and have been widely used for password research [56], [16], [30], [52], [55], [57], [51]. By combining the 10 databases, we obtain 430,145,229 unique email addresses and 349,553,965 unique usernames. We select the top 10,000 most popular usernames for our data collection. The top 5 usernames are info, john, admin, mail, and david, where "info" and "david" have been used 800,000 and 86,000 times, respectively.

To confirm that popular usernames are more likely to receive emails, we perform a quick pilot test. We scan all 7 disposable email services, and examine how many addresses under the 10,000 most popular usernames contain incoming emails. From a one-time scan, we find that 8.74% of the popular usernames contain emails *at the moment we checked the inbox*. As a comparison, we scan a list of random 10,000 usernames and found that only about 1% of addresses contain emails, which confirms our intuition.

Time Interval for Crawling. For each disposable email service, we build a crawler to periodically check the email addresses under the top 10,000 usernames. To minimize the impact on the target service, we carefully control the crawling speed and force the crawler to pause for 1 second between two consecutive requests. In addition, we keep *a single crawling thread* for each service. Under this setting, it would take more than 6 hours to scan all 10K addresses. Considering that certain disposable email services would frequently dispose incoming emails, our strategy is to have an *early timeout*. Suppose a service keeps an email for t hours, we design our crawler to stop the current scan once we hit the t-hour mark, and immediately start from the top of the username list. This strategy is to make sure we don't miss incoming emails to the most popular addresses. Since emails are more likely to hit the top addresses, this strategy allows us to collect more emails with the limited crawling speed.

To set up the early-timeout, we need to measure the email deletion time for each service. We perform a simple experiment: for each service, we first generate 25 random MD5 hash strings as usernames. This is to make sure these addresses are not accidentally accessed by other users during the experiment.

[1] Two of the RA services have adopted CAPTCHAs for their sites.

TABLE I: The expiration time of disposable emails. We show the expiration time claimed on the website and the actual expiration time obtained through measurements.

Website	Claimed Time	Actual Time (Min., Avg., Max.)
guerrillamail.com	"1 hour"	1, 1, 1 (hour)
mailinator.com	"a few hours"	10.5, 12.6, 16.5 (hours)
temp-mail.org	"25 mins"	3, 3, 3 (hours)
maildrop.cc	"Dynamic"	1, 1, 1 (day)
mailnesia.com	"Dynamic"	12.6, 12.8, 13.1 (days)
mailfall.com	"25 mins"	30, 30, 30 (days)
mailsac.com	"Dynamic"	19.9, 20.3, 20.7 (days)

Then, we send 25 emails in 5 batches (12 hours apart). In the meantime, we have a script that constantly monitors each inbox to record the message deletion time. In this way, we obtain 25 measurements for each disposable email service.

As shown in Table I, disposable email services often don't delete emails as quickly as promised. For example, `mailfall.com` claimed to delete emails in 25 minutes but in actuality, held all the emails for 30 days. Similarly `temp-mail.org` claimed to delete emails in 25 minutes but kept the emails for 3 hours. This could be an implementation error of the developers or a false advertisement by the service. Many of the services claim that the expiration time is not fixed (which depends on their available storage and email volume). Based on Table I, we only need to apply the early-timeout for `temp-mail` and `guerrillamail` to discard lower-ranked usernames, using a timeout of 1 hour and 3 hours respectively.

B. Disposable Email Dataset

We applied the crawler to 7 disposable email services from October 16, 2017 to January 16, 2018 for three months. In total, we collected 2,332,544 email messages sent to monitored email addresses. Our crawler is implemented using Selenium [7] to control a headless browser to retrieve email content. The detailed statistics are summarized in Table II. For 5 of the disposable email services, we can cover all 10K addresses and almost all of them have received at least one email. For the other 2 email services with very a short expiration time (`temp-mail` and `guerrillamail`), we focus on an abbreviated version of the popular usernames list. The number of emails per account has a highly skewed distribution. About 48% of disposable email addresses received only one email, and 5% of popular addresses received more than 100 emails each.

Each email message is characterized by an email title, email body, receiver address (disposable email address), and sender address. As shown in Table II, not all emails contain all the fields. 4 of the 7 disposable email services do not always keep the sender email addresses. Sometimes the disposable email services would intentionally or accidentally drop sender addresses. In addition, spam messages often omit the sender address in the first place. In total, there are 1,290,073 emails (55%) containing a sender address (with a total of 452,220 unique sender addresses). These sender addresses correspond to 210,373 unique sender domain names. From the email body, we extracted 13,396,757 URLs (1,031,580 unique URLs after removing URL parameters).

TABLE II: Statistics of the collected datasets.

Website	# Emails	Dispos. Address	Uniq. Sender Address (Domain)	Msgs w/ Sender Address
guerrillamail	1,098,875	1,138	410,457 (190,585)	1,091,230 (99%)
mailinator	657,634	10,000	27,740 (16,342)	55,611 (8%)
temp-mail	198,041	5,758	1,748 (1,425)	13,846 (7%)
maildrop	150,641	9,992	786 (613)	3,950 (3%)
mailnesia	106,850	9,983	1,738 (686)	4,957 (5%)
mailfall	75,179	9,731	3,130 (288)	75,164 (100%)
mailsac	45,324	9,987	11,469 (8,019)	45,315 (100%)
Total	2,332,544	56,589	452,220 (210,373)	1,290,073 (55%)

Biases of the Dataset. This dataset provides a rare opportunity to study disposable email services and email tracking. However, given the data collection method, the dataset inevitably suffers from biases. We want to clarify these biases upfront to provide a more accurate interpretation of the analysis results later. First, our dataset only covers the user-specified addresses but not the randomly-assigned addresses. Second, our data collection is complete with respect to the popular email addresses we monitored, but is incomplete with respect to all the available addresses. As such, any "volume" metrics can only serve as a lower bound. Third, we don't claim the email dataset is a representative sample of a "personal inbox". Intuitively, users (in theory) would use disposable email addresses differently relative to their personal email addresses. Instead, we argue the unique value of this dataset is that it covers a wide range of *online services* that act as the email senders. The data allows us to empirically study email tracking from the perspective of online services (instead of the perspective of email users). It has been extremely difficult (both technically and ethically) for researchers to access and analyze the email messages in users' personal inboxes. Our dataset, obtained from public email gateways, allows us to take a first step measuring the email tracking ecosystem.

C. Ethical Considerations and IRB

We are aware of the sensitivity of the dataset and have taken active steps to ensure research ethics: (1) We worked closely with IRB to design the study. Our study was reviewed by IRB and received an exemption. (2) Our data collection methodology is designed following a prior research study on disposable SMS services [41]. Like previous researchers, we carefully have controlled the crawling rate to minimize the impact on the respective services. For example, we enforce a 1-second break between queries and explicitly use a single-thread crawler for each service. (3) All the messages sent to the gateways are publicly available to any Internet users. Users are typically informed that other users can also view the emails sent to these addresses. (4) We have spent extensive efforts on detecting and removing PII and personal emails from our dataset (details in §IV-A). (5) After data collection, we made extra efforts to reach out to users and offer users the opportunity to opt out. More specifically, we send out an email to each of the disposable email addresses in our dataset, to inform users of our research activity. We explained the purpose of our research and offered the opportunity for users to withdraw their data. So far, we did not receive any data withdraw request. (6) Throughout our analysis, we did not attempt to analyze or access any individual accounts registered under the disposable email addresses. We also did not attempt to click on any URLs in the email body (except the automatically loaded tracking pixels). (7) The dataset is stored on a local server with strict access control. We keep the dataset strictly to ourselves.

Overall, we believe the analysis results will benefit the community with a deeper understanding of disposable email services and email tracking, and inform better security practices. We hope the results can also raise the awareness of the risks of sending sensitive information over public channels.

IV. Analyzing Disposable Emails

In this section, we analyze the collected data to understand how disposable email services are used in practice. Before our analysis, we first detect and remove PII and the potential personal emails from the dataset. Then we classify emails into different types and infer their use cases. More specifically, we want to understand what types of online services with which users would register. Further, we seek to understand how likely it is for disposable email services to be used in sensitive tasks such as password resets.

A. Removing PII and Personal Emails

Removing PII. Since email messages sent to these gateways are public, we suspect careless users may accidentally reveal their PII. Thus, we apply well-established methods to detect and remove the sensitive PII from the email content [49]. Removing PII upfront allows us to analyze the dataset (including manual examination) without worrying about accidentally browsing sensitive user information. Here, we briefly introduce the high-level methodology and refer interested readers to [49] for details. The idea is to build a list of regular expressions for different PII. We first compile a ground-truth dataset to derive regular expressions and rules. Like [49], we also use the public Enron Email Dataset [8] which contains 500K emails. We focused on the most sensitive PIIs and labeled a small ground-truth set for credit card numbers, social security numbers (SSN), employer identification numbers (EIN), phone numbers, and vehicle identification numbers (VIN) as shown in Table III. Then we build regular expressions for each PII type. For credit card numbers, we check the prefix for popular credit card issuers such as VISA, Mastercard, Discover and American Express, and we also use Luhn algorithm [32] to check the validity of a credit card number. As shown in Table III, the regular expressions have good precision and recall.

We applied the regular expressions to our dataset and detected a large number of PIIs including 1,399 credit card numbers, 926 SSNs, 701 EINs, and 40K VINs and 700K phone numbers. All the detected PII are automatically blacked-out by the scripts. Note that the 700K phone numbers are not necessarily users' personal phone numbers, but can be phone numbers of the email sending services. We take a conservative approach to blackout all the potential PII. The results indicate that people indeed use the disposable email services to communicate sensitive information.

TABLE III: PII detection accuracy based on ground-truth, and the number of detected PII instances in our dataset.

PII	Ground-truth Evaluation					# Detected in
Type	#Email	#Inst.	F1	Precis.	Recall	Our Data
Credit	16	25	1.00	1.00	1.00	1,399
SSN	13	15	1.00	1.00	1.00	926
EIN	16	29	1.00	1.00	1.00	701
Phone	20	50	0.99	0.98	1.00	726,138
VIN	15	19	0.97	1.00	0.95	43,438

Removing Personal Emails. We further remove potentially personal emails including replied emails and forwarded emails. We filter these emails based on "Re: " and "Fwd: " in the email titles. Although this step may not be complete, it helps to delete email conversations initiated by the users. In total, we filter out 30,955 such emails (1.33%). This again shows use of disposable email addresses for personal communications.

B. Categorizing Disposable Emails

Next, using the remaining data, we infer the common use cases of disposable email services by classifying email messages. First, we manually analyze a sample of emails to extract the high-level categories of emails (*ground-truth* dataset). Second, we build a machine learning classifier and use it to classify the unlabeled emails. Third, we analyze the classification results to examine common usage cases.

Manual Analysis and Email Clustering. To assist the manual analysis, we first cluster similar email messages together. For efficiency considerations, we only consider the *subject* (or title) of the email message for the clustering. Since we don't know the number of clusters in the dataset, we exclude clustering methods that require pre-defining the number of clusters (*e.g.*, K-means). Instead, we use ISODATA algorithm [13] which groups data points based on a cut-off threshold of the similarity metric. We use Jaccard index to measure the keyword similarity of two email subjects. Given two email subjects, we extract all their keywords into two sets $\mathbf{w_i}$ and $\mathbf{w_j}$. Then we calculate their similarity as $sim(i,j) = \frac{|\mathbf{w_i} \cap \mathbf{w_j}|}{|\mathbf{w_i} \cup \mathbf{w_j}|}$.

We set the cut-off threshold as 0.2 to loosely group similar email titles together. In total, we obtain 91,306 clusters, most of which are small with less than 100 emails (98%). The cluster size distribution is highly skewed. The top 500 clusters cover 56.7% of the total email messages. A few large clusters (with over 1000 emails) typically represent spam campaigns. To make sure 0.2 is a reasonable threshold, we have tried even smaller thresholds to merge some of the clusters. For example, if we set the threshold to 0.1 and 0.01, we get 26,967 and 19,617 clusters respectively. However, manual examination shows that the emails in the same cluster no longer represent a meaningful group. We stick to 0.2 as the threshold. By manually examining 500+ clusters (prioritizing larger ones), we summarize 4 major types of emails.

- **Account Registration**: emails to confirm account registration in online services.
- **Password Reset**: emails that instruct the user to reset passwords for an online account.
- **Authentication**: emails that contain a one-time authentication code for login.
- **Spam**: all other unsolicited emails including newsletters, advertisements, notifications from online services, and phishing emails.

Email Classification. We need to further develop an email classifier because the clusters do not map well to each of the email categories. For example, a cluster may contain both spam emails and emails that are used to confirm account registration. Below, we build a machine learning classifier to classify emails into the four categories.

For classifier training, we manually labeled a ground-truth dataset of 5,362 emails which contains 346 account registration emails, 303 password reset emails, 349 authentication emails and 4,364 spam emails. Note that we have labeled more spam emails than other categories because our manual examination suggests that there are significantly more spam emails in the dataset. For each email, we combine the text in the email title and the email body, and apply RAKE (Rapid Automatic Keyword Extraction) [44] to extract a list of keywords. RAKE is a *domain independent* keyword extraction algorithm based on the frequency of word appearance and its co-occurrence with other words. In this way, less distinguishing words such as stopwords are automatically ignored. We use extracted keywords as features to build a *multi-class* SVM classifier. We have tested other algorithms such as Decision Tree and Random Forests. However, the SVM performed the best. We also tested word2vector [35] to build the feature vector, and its results are not as good as RAKE (omitted for brevity).

Through 5-fold cross-validation, we obtain a precision of 97.23% and a recall of 95.46%. This is already highly accurate for a *multi-class* classifier — as a baseline, a random classification over 4 classes would return an accuracy of 25%. We manually checked some of the classification errors, and found that a few account registration and authentication emails are labeled as spam due to “spammy” keywords (*e.g.*, “purchase”).

Note that two types of emails are not applicable here. First, 58,291 (2.50%) of the emails do not have any text content. Second, 535,792 (22.97%) emails are not written in English. Since our classifier cannot analyze the text of these emails, they are not part of the classification results in Figure 2 (we still consider these emails in the later analysis of email tracking). To make sure our classification results are trustworthy, we randomly sampled 120 emails (30 per category) to examine manually. We only find 5 misclassified emails (4% error rate), which shows that the ground-truth accuracy transfers well onto the whole dataset.

C. Inferring Usage Cases

Next, we examine disposable email service usage. Recall that our dataset contains emails *received* by the disposable email addresses. Intuitively, after the users obtain the disposable email addresses, they will use the email addresses for certain online tasks (*e.g.*, registering accounts), which will expose the addresses and attract incoming emails. By analyzing these incoming emails, we can infer at which services the user registered the accounts, and what the accounts are used for.

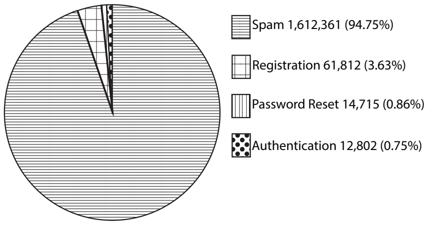


Fig. 2: Email classification results.

Types of Emails. As shown in Figure 2, while spam emails take the majority, there is a non-trivial number of emails that are related to account management in various online services. In total, there are 89,329 emails involved with account registration, password resets or sending authentication codes. These emails are sent from 168,848 unique web domains. We refer these 3 types of emails as *account management emails.* Account management emails are indicators of previous interactions between the user and the email sending domain. They are explicit evidence that users have used the disposable email addresses to register accounts in the web services.

Breakdown of Spam Emails. The spam emails take a large portion of our dataset (1,612,361 emails, 94%), which deserve a more detailed break-down. Some of the spam messages also indicate previous interactions between a user and the email sender. For example, if a user has registered an account or RSS at an online service (*e.g.* Facebook), this service may periodically send “social media updates”, “promotions”, or notifications to the disposable email address. We call them *notification spam*. Such notification messages almost always include an *unsubscribe* link at the bottom of the email to allow users to opt out. As such, we use this feature to scan the spam messages and find 749,602 notification messages (counting for 46.5% of the spam messages).

The rest of *unsolicited* spam messages may come from malicious parties, representing malware or phishing campaigns. To identify the malicious ones, we extract all the clickable URLs from the email content, and run them against the VirusTotal blacklists (which contains over 60 blacklists maintained by different security vendors [41], [11]), and the eCrimeX blacklist (a phishing blacklist maintained by the Anti Phishing Work Group). In total, we identify 84,574 malicious spam emails (5.2%) that contain at least one blacklisted URL.

Finally, we apply the same ISODATA clustering algorithm to the rest of the spam emails (which count for 48.3%) to identify spam campaigns. We find 19,314 clusters and the top 500 clusters account for 75.6% of the spam emails. Manual examination shows that the top clusters indeed represent

TABLE IV: Top 5 sender domains of registration emails, password reset emails and authentication emails.

Rk.	Registration Emails			Password Reset Emails			Authentication Emails		
	sender domain	# msg	category	sender domain	# msg	category	sender domain	# msg	category
1	facebookmail.com	2,076	Social Net	facebookmail.com	931	Social Net	frys.com	987	Shopping
2	gmail.com	1,015	Webmail	twitter.com	508	Social Net	paypal.com	622	Business
3	aol.com	928	Search	miniclip.com	415	Games	ssl.com	418	IT
4	avendata.com	733	Business	retailio.in	223	Business	id.com	163	Business
5	axway.com	720	Education	gmail.com	145	Webmail	facebookmail.com	161	Social Net

TABLE V: Top 10 categories of the email sender domains for spam and account management emails.

Rk.	Account Management Email		Spam Email	
	Category	# Msg (domain)	Category	# Msg (domain)
1	Business	12,699 (2,079)	Business	251,822 (31,433)
2	IT	6,759 (1,228)	Marketing	145,538 (1,855)
3	Software	5,481 (571)	IT	108,933 (6,091)
4	Social Net	5,362 (149)	Shopping	104,361 (5,361)
5	Marketing	5,320 (430)	Social Net	102,342 (1,223)
6	Shopping	3,307 (370)	Education	73,038 (6,218)
7	Education	2,946 (673)	Software	44,560 (3,217)
8	Search	2,154 (74)	Travel	39,211 (3,444)
9	Finance	2,017 (302)	News	38,567 (1,533)
10	Webmail	1,575 (46)	Adult	30,777 (1,344)

large spam campaigns, most of which are pornography and pharmaceutical spam.

Categories of Email Senders. To understand what types of online services users interact with, we further examine the "categories" of email sender domains. The "categories" are provided by VirusTotal. Table V shows the top 10 categories for spam emails and account management emails. We have two main observations.

First, the emails are sent from a very broad range of domain categories. This suggests that users have used the disposable email addresses to register accounts in all different types of websites. There are in total 121 different categories, and the top-10 categories only cover 51.01% of account management emails and 58.25% of spam emails, which confirms the high diversity of usage. Second, we observe that disposable email addresses are often used to register potentially *sensitive* accounts. Categories such as "online social networks", "finance", "shopping" have made the top-10 for account management emails. This could introduce risks if a user accidentally left PII or credit card information in the registered account. Accounts registered under disposable email addresses are easily hijackable. Any other users can take over the registered accounts by sending a password-reset link to the disposable email address, which will be publicly accessible. Given the 14,000+ password-reset emails in our dataset, it is possible that malicious parties are already performing hijacking.

Case Studies: Common Usage. Next, we use specific examples to illustrate the common usage cases. Table IV lists the top 5 email sending domains for registration, password reset and authentication emails. We show users use disposable email addresses to register accounts in gaming and social network services in order to enjoy the online services without giving away real email addresses. For example, facebookmail.com appears in the top-5 of all three types of emails. twitter and miniclip (for gaming) also fall into the same category. It is possible that some accounts are fake accounts registered by spammers [58]. Since we decided not to back-track (or login into) any individual user's account for ethical considerations, we cannot systematically differentiate them. Previous research on anonymous community (*e.g.*, 4chan, Reddit) show that users prefer anonymized identifiers when posting sensitive or controversial content [54], [33]. We suspect normal users may use the disposable email address to create such social media accounts for similar purposes. PayPal accounts have additional risks. If a user accidentally binds a real credit card to the account, it means any other users may take over the PayPal account by resetting the password.

Another common use case is to obtain *free goods*. For example, users often need to register an email address to obtain demos or documents from software solutions and educational services, *e.g.*, axway.com, avendata.com, retailio.in, and ssl.com. Users can also obtain a discount code from shopping services (*e.g.*, frys.com). Another common case (not in the top-5) is to use the disposable email address to register for free WiFi in airports and hotels. Finally, we observe cases (not in the top 5) where users try to preserve *anonymity*: For example, people used disposable email addresses to file anonymous complaints to the United States Senate (86 emails).

Note that gmail.com is special: it turns out that many small businesses cannot afford their own email domains and directly use Gmail (*e.g.*, pizza@gmail.com). Thus, The domain gmail.com does not represent Gmail, but is a collection of small businesses. aol.com has a similar situation.

Case Studies: Risky Usage. We observe other cases that may involve risks. These cases may be not as common as those shown in Table IV, but if their accounts are hijacked (through the public disposable email addresses), the real-world consequences are more serious. For example, there are 4,000+ emails from *healthcare.gov*, the website of the Affordable Care Act. It is likely that people have used disposable email addresses to register their healthcare accounts where each account carries sensitive information about the user.

Similarly, there are emails from mypersmail.af.mil (Air Force Service Center), suggesting that people have used disposable email address to register Air Force personnel accounts. The registration is open to civilian employees who must use their SSN and date of birth for the registration [1]. A password reset option is also available on the website.

In addition, more than 32,990 emails are used to receive *scanned documents* from PDF scanning apps (*e.g.*, Tiny Scanner). It is possible for an attacker to obtain all the scanned documents by hijacking these disposable email addresses.

Finally, there are over 1000 emails from *digital currency* or digital wallet services such as buyabitcoin.com.au and thebillioncoin.info. While most emails are related to account registrations, some are related to bitcoin purchase confirmations (*e.g.*, receipts). If these accounts hold bitcoins, anyone has a chance to steal them.

D. Summary

We show that disposable email services are primarily used to register online accounts. While most of the incoming emails are spam and notifications (94%), we did find a large number of emails (89,000+) that are related to account registration, password reset, and login authentication. There is a strong evidence that users use disposable email services for sensitive tasks. We find 1000+ credit card numbers and 926 SSNs accidentally revealed in the emails and 30K replied and forwarded emails that indicate a personal usage. More importantly, accounts registered with disposable email addresses can be easily hijacked through a password reset.

V. Email Tracking Measurements

Next, we use the large-scale email dataset to analyze email tracking in the wild. We seek to answer three key questions. First, what types of tracking techniques do trackers use in practice, and what is the nature of the data leaked through tracking. Second, how prevalent is third-party tracking among different types of online services? Third, who are the top trackers in the tracking ecosystem and how dominant are they? In the following, we first describe the threat model and our method to detect third-party tracking, and then present the measurement results.

A. Threat Model

By embedding a small image in the email body, the email sender or third-parties can know whether the email has been opened by the receiver. When an email is opened, the tracking pixel will be *automatically* loaded from a remote server via HTTP/HTTPS (which does not require any user actions). Based on the request, the remote server will know who (based on the email address or other identifiers) opened the email at what location (based on IP) and what time (timestamp) using what device ("User-Agent"). The privacy leakage is more serious when the remote server is a third-party.

Email tracking works only if the user's email client accepts HTML-based email content, which is true for most modern email clients. However, careful users may use ad-blockers to block tracking pixels [17]. In this paper, we make no assumption about a user's email client, and only focus on the tracking content in the email body. Note that JavaScript is not relevant to email tracking since JavaScript will not be automatically executed [4]. Alternatively, email tracking can be done through querying font files. We did not find any font-based tracking in our dataset and omit it from the threat model.

B. Tracking Detection Method

Given an email, we design a method to determine if the email contains tracking pixels. First, we survey popular email tracking services (selected through Google searching) to examine how they implement the tracking pixels. After analyzing Yesware, Contact Monkey, Mailtrack, Bananatag, Streak, MailTracker, The Top Inbox, and Hub Spot, we observe two common characteristics. First, all 8 services embed small or transparent HTML image tags that are not visible to users (to remain stealthy). Second, the image URLs often contain some form of user identifiers (either the receiver's email address or IDs created by the tracking services). This is because the tracker wants to know "who" opened the email. Next, we design a detection method based on these observations.

Steps to Detect Pixel Tracking. Given an email, we first extract all the HTML image tags and corresponding URLs. Here, we focus on tracking URLs that notify the tracker about the user identity. We filter out links that do not contain any parameters[2]. Then for each image URL, we follow the four steps below to detect email tracking.

- **Step 1: Plaintext Tracking Pixel**: if the link's parameters contain the receiver's email address in plaintext, then the image is a tracking pixel.
- **Step 2: Obfuscated Tracking Pixel**: if the link's parameters contain the "*obfuscated version*" of the receiver's email address, then the image is a tracking pixel. We apply 31 hash/encoding functions on the receiver email address to look for a match (see Appendix). We also test two-layer obfuscations by exhaustively applying two-function combinations, *e.g.*, MD5(SHA1()). In total, we examine 992 obfuscated strings for each address. We didn't consider salted obfuscation here due to the extremely high testing complexity.
- **Step 3: Invisible HTML Pixel**: we check if the image is trying to hide based on the HTML height and width attributes. We consider the image as a tracking pixel if both the height and width are below a threshold t or the HTML tag is set to be "hidden" or "invisible".
- **Step 4: Invisible Remote Pixel**: trackers may intentionally set a large height or width in HTML to avoid detection. If the HTML height or width is above t, we use a web crawler to fetch the actual image from the remote server. If the actual image size is below t, regardless the HTML attributes, we regard it as a tracking pixel.

Step-1 and step-2 are adapted from the method described in [17]. We explicitly look for parameters in the image URL that leak the receiver's email address. However, it is still possible that trackers use an obfuscation method that is not listed in Table XI (*e.g.*, keyed-hash). More importantly, the tracker can use a random string as the identifier and keep the mapping in the back-end. As such, we introduce step 3 and step 4 as a complementary way to capture the tracking behavior that cannot be detected by [17].

[2] Image URLs without parameters will still reveal the user's IP but are not necessarily for tracking

TABLE VI: Email tracking detection results. *Tracking party is based on 1.29 million emails that have a sender address.

Attributes	Total	Tracking Stats	Tracking Party*		Tracking Method			
			1st-party	3rd-party	Plaintext	Obfuscat.	Invis. HTML	Invis. remote
# Image URLs	3,887,658	1,222,961 (31.5%)	509,419	179,223	200,682	200,247	548,166	537,266
# Email Messages	2,332,544	573,244 (24.6%)	264,501	149,303	35,702	29,445	473,723	124,900
# Sender Domains	210,373	11,688 (5.5%)	5,403	7,398	1,478	597	9,149	1,802
# Tracker Domains	N/A	13,563	5,381	2,302	2,403	984	9,935	2,282

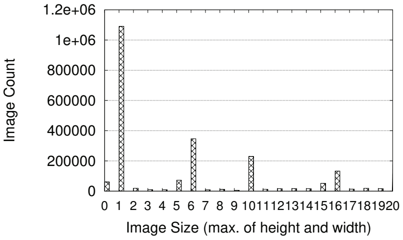


Fig. 3: Distribution of the HTML image size.

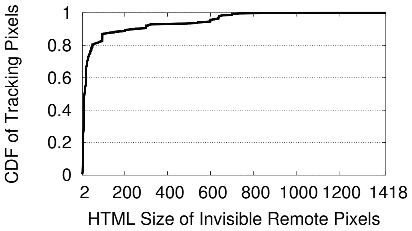


Fig. 4: The HTML image size of invisible remote pixels.

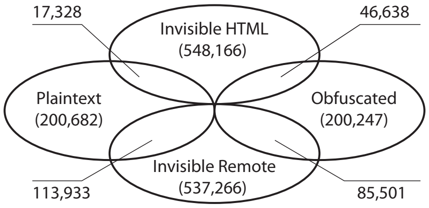


Fig. 5: # of tracking URLs under different tracking methods.

To set the threshold t for tracking pixels, we plot Figure 3 to show the image size distribution in our dataset. Image size is defined as the larger value between the height and width. As shown in Figure 3, there is a clear peak where the image size is 1 (1.1 million images). There are also 60K images of a "zero" size. To be conservative, we set the threshold $t = 1$. Our method is still not perfect, since we might miss trackers that use bigger tracking images. The detection result is only a lower-bound of all possible tracking.

Alternative Tracking Methods. In addition to the methods above, we have tested other alternative methods, which did not return positive results in our pilot test. For completeness, we briefly discuss them too. First, other than URL parameters, trackers use subdomain names to carry the user identifiers. For example, a tracker (*e.g.*, `tracker.com`) may register many subdomains, and use each subdomain to represent a user (e.g., `u1.tracker.com`, `u2.tracker.com`). To look for such trackers, we sort the domain names of image URLs based on their number of subdomains. We only find 3 domain names (list-manage.com, sendgrid.com and emltrk.com) that have more than 1000 subdomains. However, we find that they are not using subdomain names as user identifiers. Instead, each subdomain is assigned to represent a "customer" that adopted their tracking services. For example, a tracking URL `office-artist.us12.list-manage.com` is used by online service `office-artist.com` to track their users. We have examined all the tracking domains with over 50 subdomains and did not find any subdomain-based tracking.

A limitation of step-1 and step-2 is that they cannot capture trackers that use a random string as the identifier. An alternative approach is cluster image URLs that follow the same templates. Then the differences in the URLs are likely to be the unique user identifiers. However, our pilot test shows that the majority of the differences in image URLs are indeed personalized content, but the personalized content is not for tracking. For example, online services often send product recommendations using the same template but use different "ProductIDs" in the image URLs. This approach easily introduces false positives.

TABLE VII: Obfuscation methods used in the tracking URLs.

1-layer Obf.	Track URLs	2-layer Obf.	Track URLs
MD5	183,527 (91.7%)	Base64 (Urlencode)	765 (0.4%)
Base64	9,876 (4.9%)	Urlencode (Base64)	134 (0.1%)
SHA1	2,754 (1.4%)	Base64 (Base64)	49 (0.0%)
Urlencode	2,094 (1.0%)	MD5 (MD5)	29 (0.0%)
Crc32	704 (0.4%)	Urlencode (Urlencode)	9 (0.0%)
SHA256	268 (0.1%)		
Base16	38 (0.0%)		

Third-party Tracking. To differentiate first-party and third-party tracking, we match the domain name of the *email sender* and that of the *image URL*. Since we use domain name to perform the matching, all the "subdomains" belong to the same party. For example, `mail.A.com` and `image.A.com` match with each other since they share the same domain name. If the email sender's domain name is different from that of the image tracking URL, we then check their WHOIS record to make sure the two domains are not owned by the same organization. We regard the tracking as a third-party tracking if the two domain names belong to different organizations.

VI. Measurement Results

We apply our detection method to the 2.3 million emails, and the results are summarized in Table VI. In total, we extracted 3.9 million unique image URLs and 1.2 million of them (31.5%) are identified as tracking links. These tracking links are embedded in 573K emails (24.6%). Out of the 210K email sender domains, we find that 11.6K of them (5.5%) have embedded the tracking pixels in their emails. In total, we identify 13,563 unique tracker domains. In the following, we first characterize different email tracking techniques and the "hidden trackers". Then we focus on third-party tracking and identify the top trackers. Finally, we analyze how different online services perform tracking.

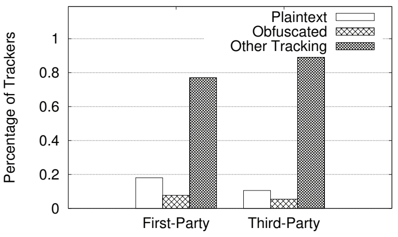


Fig. 6: Different tracking methods of first-party and third-party trackers.

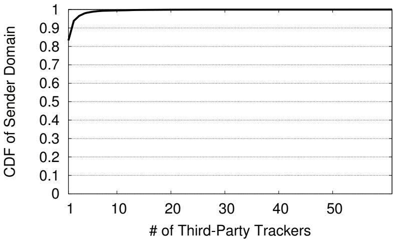


Fig. 7: # of third-party trackers per sender.

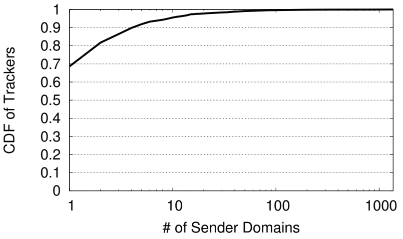


Fig. 8: # of sender domains associated to each tracker.

A. Email Tracking Techniques

As shown in Table VI, there is almost an equal number of tracking URLs that send plaintext user identifiers (200,682) and those that send obfuscated identifiers (200,247). For the obfuscated tracking, we find 12 obfuscated methods are used by trackers (out of 992 obfuscations tested). As shown in Table VII, MD5 is applied in the vast majority of these tracking URLs (91.7%) followed by Base64 (4.9%). We did find cases where the obfuscation functions are applied more than once but these cases are rare (<0.5%). This observation is consistent with the previous smaller-scale study [17].

There are even more tracking links that use invisible pixels. We find 548,166 invisible *HTML* pixels where the HTML size attributes are 1×1 or smaller or the image tags are set to be “hidden”. Meanwhile, we find 537,266 additional invisible *remote* pixels which falsely claim their HTML size attributes even though the actual image is 1×1. By analyzing the HTML attributes of the invisible remote pixels, we find that 20% of them did not specify the size attributes. For the remaining images that specified the size, Figure 4 shows the size distribution. These pixels declare much larger image sizes in HTML (possibly to avoid detection) while the actual image is only 1×1 (invisible to users).

Figure 5 shows the overlaps of the tracking URLs detected by different methods. We find 17K (8.6%) the plaintext tracking URLs are also using invisible HTML pixels; 114K (56.8%) plaintext tracking URLs are using invisible remote pixels. This suggests that trackers prefer stealthier methods when sending plaintext identifiers. For obfuscated tracking URLs, although the “remote” invisible pixels are still preferred (86K, 42.7%), the ratio is more balanced compared to the usage of HTML pixels (47K, 23.3%). When the parameters are obfuscated, the trackers are likely to put in less effort towards hiding their tracking pixels.

TABLE VIII: Top 10 hidden trackers, ranked by the # of trackers that redirect traffic to them.

Rank	Hidden Tracker	# Direct Trackers	# Emails
1	liadm.com	252	29,643
2	scorecardresearch.com	227	27,301
3	eloqua.com	192	3,639
4	doubleclick.net	164	96,430
5	rlcdn.com	132	42,745
6	adsrvr.org	130	48,858
7	pippio.com	59	41,140
8	hubspot.com	47	3,995
9	serving-sys.com	41	18,116
10	dotomi.com	40	23,526

Hidden Trackers. Through our analysis, we find *hidden trackers* when we try to fetch the tracking pixels from the remote servers. More specifically, when we request the images, the request will be first sent to the “direct tracker” (based on the image URL) and then redirected to the “hidden trackers”. The hidden trackers are not directly visible in the email body and can only be reached through HTTP/HTTPS redirections. In this way, user identifiers are not only leaked to the direct tracker but also to the hidden trackers in real time. Intuitively, hidden trackers are less likely to be blacklisted (by adblockers) since they do not directly appear in the HTML. To capture hidden trackers, we crawled all of the 1,222,961 tracking URLs. We find that a large number of the tracking URLs have redirections (616,535, 50.4%). In total, we obtain 2,825 unique hidden tracker domains. Table VIII shows the top 10 hidden trackers (ranked by the number of the direct trackers that redirect traffic to them).

Hidden trackers may also act as direct trackers in certain emails. We find that 2,607 hidden trackers have once appeared to be direct trackers in out dataset. In total, hidden trackers are associated with 112,068 emails and 2260 sender domains (19.3% of sender domains that adopted tracking). Interestingly, many *first-party* tracking links also share the user information with hidden trackers in real-time. More specifically, there are 9,553 emails (220 sender domains) that share user identifiers to a hidden tracker while performing first-party tracking.

B. Third-party Tracking

Next, we focus on third-party tracking and identify the top trackers. This analysis is only applicable to emails that contain a sender address (1.2 million emails).

Overall Statistics. Third-party tracking is highly prevalent. As shown in Table VI, there are 149k emails with third-party tracking. Interestingly, there are more sender domains with third-party tracking (7,398) than those with first-party tracking (5,403). In total, we identify 2,302 third-party trackers.

Figure 6 breaks-down the tracking methods used by first- and third-party trackers. To make sure different tracking methods don’t overlap, we present plaintext tracking and obfuscated tracking, and regard the rest of the invisible pixel tracking as

TABLE IX: Top third-party trackers for each type of tracking method.

Rk.	Top Trackers (# Sender Domains / # Email Messages)			
	plaintext (total: 513 / 4,783)	obfuscated (total: 200 / 5,737)	invis. HTML (total: 6,106 / 126,286)	invis. remote (total: 1,180 / 21,906)
1	mczany.com (66 / 290)	alcmpn.com (36 / 2,173)	list-manage.com (1,367 / 19,564)	hubspot.com (168 / 743)
2	emltrk.com (61 / 956)	pippio.com (29 / 2,104)	sendgrid.net (849 / 10,416)	google-analytics.com (164 / 3,671)
3	socursos.net (28 / 93)	rlcdn.com (11 / 246)	returnpath.net (333 / 12,628)	rs6.net (98 / 629)
4	vishalpublicschool.com (27 / 65)	dotomi.com (11 / 218)	rs6.net (217 / 2645)	doubleclick.net (56 / 2,678)
5	52slots.com (26 / 48)	bluekai.com (8 / 201)	emltrk.com (197 / 2,362)	tradedoubler.com (29 / 98)
6	joyfm.vn (18 / 26)	emailstudio.co.in (6 / 17)	klaviyomail.com (112 / 2,188)	mixpanel.com (29 / 144)
7	jiepop.com (17 / 52)	acxiom-online.com (5 / 517)	exct.net (103 / 491)	salesforce.com (27 / 64)
8	karacaserigrafi.com (16 / 120)	lijit.com (5 / 118)	exacttarget.com (88 / 2,203)	publicidees.com (15 / 84)
9	dfimage.com (15 / 53)	sparkpostmail.com (5 / 9)	dripemail2.com (86 / 919)	gstatic.com (14 / 191)
10	doseofme.com (15 / 32)	mmtro.com (4 / 85)	adform.net (76 / 550)	mfytracker.com (12 / 16)

TABLE X: Top third-party trackers across the full dataset. “⭘” means the tracker is also a hidden tracker. “○” means the tracker is not a hidden tracker.

Rk.	Top Trackers	Type	# Senders	# Emails
1	list-manage.com	○	1,367	19,564
2	sendgrid.net	○	849	10,416
3	returnpath.net	○	345	12,784
4	rs6.net	○	292	3,274
5	emltrk.com	○	226	3,328
6	google-analytics.com	○	225	5,174
7	doubleclick.net	⭘	208	12,968
8	hubspot.com	⭘	192	874
9	eloqua.com	⭘	150	1,981
10	rlcdn.com	⭘	133	7,117
Subtotal			3,715 (31.8%)	68,914 (12.0%)

“other tracking”. Figure 6 shows that third-party trackers are less likely to collect the user email address as the identifier.

Figure 7 shows the number of third-party trackers used by each sender domain (corresponding to an online service). We find that the vast majority (83%) of online services use a single third-party tracker. About 17% of online services have multiple third-party trackers, sharing user information with multiple-parties at the same time. The extreme case is `amazonses.com` which uses 61 third-party trackers.

Top Trackers. From the third-party tracker's perspective, Figure 8 shows that only a small number of trackers are used broadly by different online services. To analyze the top trackers, we present Table IX to list top third-party trackers for each tracking method. We rank the trackers based on the number of online services that use them. A popular tracker should be used by many online services. For reference, we also show the number of emails associated with each tracker.

We observe that top trackers under different tracking methods rarely overlap with each other. This indicates that a tracker usually sticks to a specific tracking method. The most dominating trackers per category are `mczany.com` (plaintext tracking), `alcmpn.com` (obfuscated tracking), `list-manage.com` (invisible HTML), and `hubspot.com` (invisible remote). Noticeably, under the “stealthy” remote tracking, we also find that `google-analytics.com` and `doubleclick.net` make the top 10, which are Google's trackers that have dominated web tracking [48], [9], [29].

Table X shows the top trackers across the full dataset, including all the hidden trackers. We show that the top 10 trackers collectively cover 33.5% of online services, and are responsible for 12% of the tracking emails. Although top trackers are taking a big share of the market, they are not as dominating as the top tracker (*i.e.* Google) in *web tracking*. For example, previous measurements show that Google can track users across nearly 80% of the top 1 million sites [31]. Clearly, in the email tracking market, Google is not yet as dominating as it is in the web tracking.

C. Tracking by Online Services

Finally, we analyze different online services and seek to understand whether the *popularity* of online services and the *service type* would correlate to different tracking behaviors.

Popular vs. non-Popular Online Services. We first examine how tracking correlates with the popularity of online services. We reference Alexa's top 1 million domains for the ranking [2]. Note that Alexa's ranking is primarily applied to the web domain instead of the email domain. Accordingly, we check the MX record of Alexa top 1 million domains to perform the match. We find that out of the 210,373 sender domains, 18,461 domains are within Alexa top 1 million, and 2,052 are within the Alexa top 10K. For our analysis, we treat the Alexa top 10K as the *popular* domains, and the rest as *non-popular* domains. In total, the small portion of popular domains (0.98%) contributed 15.9% of the total emails.

Figure 9 shows that tracking is much more prevalent among popular domains. About 50% of popular domains adopted tracking in their emails. As a comparison, less than 10% of non-popular domains have adopted email tracking. Regarding different tracking methods, plaintext tracking and obfuscated tracking are not as prevalent as invisible pixel tracking, which is true for both popular and non-popular domains. Figure 10 shows that popular domains are slightly more likely to have first-party tracking than third-party tracking. Figure 11 shows that popular domains are more likely to use tracking methods that are harder to detect. More specifically, we focus on two types of stealthy tracking including: *invisible remote pixels* (where the HTML tags falsely claim the image size) and *hidden trackers* (trackers hide behind redirection). We observe a big difference: about 12% – 16% of popular domains have used stealthy tracking and only 1% of non-popular domains use such tracking methods.

Type of Online Services. In Figure 12, we focus on the top 10 categories of sender domains and analyze the ratio of them that adopted email tracking. Not too surprisingly, “marketing”

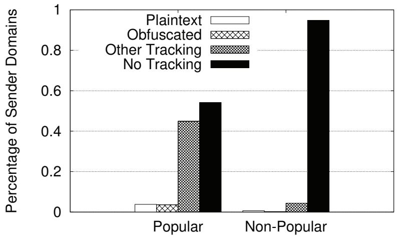


Fig. 9: Tracking methods used by popular (Alexa top 10K) and non-popular sender domains.

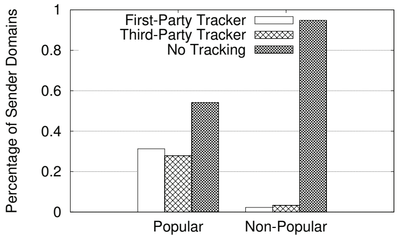


Fig. 10: Tracking methods used by popular (Alexa top 10K) and non-popular sender domains.

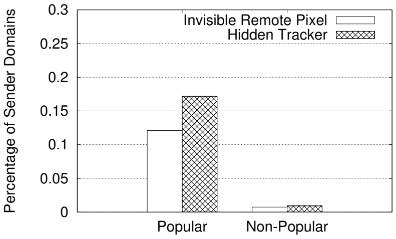


Fig. 11: Evasion methods used by popular (Alexa top 10k) and non-popular sender domains.

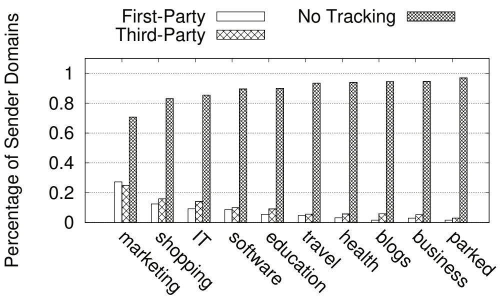


Fig. 12: Type of tracking used by different sender domains.

services have the highest ratio of tracking. In fact, many *marketing services* themselves are email tracking services (first-party tracking). Popular tracking domains also include shopping websites and information technology websites.

VII. Discussion

Risk Mitigation for Disposable Email Addresses. Our study reveals risky use cases of disposable email services. The root source of risk is the public nature of the disposable email inboxes. Randomly-assigned addresses cannot fully mitigate this problem since multiple users can still access the same address at the same time (see §III-A). One possible countermeasure is to implement sandbox using cookies. For example, if a current user is using the inbox, then other users who do not possess the same cookie cannot access the same inbox. The inbox will become available again once the current user closes her session. If the disposable email service does not implement sandbox, we believe it is necessary for the service to clearly inform users about the *public* nature of the inbox. In addition, it is also important for the service to clearly communicate the *email expiration time* to users. Our results show that two disposable email services host the emails much longer than what they promised (*e.g.*, 30 days of delay).

Users of disposable email services should proactively delete their emails whenever possible. More importantly, users should avoid revealing their PII in both the temporary inbox and in the accounts they registered through the disposable email address. Due to the public nature of the disposable email services, accounts registered with disposable email addresses can be easily hijacked through a password reset. A future direction is to understand user perceptions towards the benefits and risks of using disposable email services and identify the potential misunderstandings with respect to their security.

Email Tracking and Countermeasures. The most straightforward way to prevent email tracking is to stop rendering emails in HTML (*i.e.*, plaintext email) or block all the outgoing requests that are not initiated by user clicks. The drawback, however, is a degradation of user experience since the images in the email (if they are not embedded) cannot be displayed. To address this problem, Gmail has a special design where the Gmail server fetches all the images on behalf of the users. In this way, the tracker cannot collect users' IP addresses. However, the tracker can still obtain the following information: (1) the user indeed opens the email; (2) the time of email opening; and (3) the user's identifier (if the identifier is a parameter of the tracking URL).

A more promising way is to perform targeted HTML filtering [17] to remove tracking related image tags. Since most of tracking pixels are invisible, removing them would not hurt the user experience. This is very similar to ad-blocking where the ad-blocker construct filtering rules to detect and remove ads on websites. In addition to static HTML analysis, we believe dynamic analysis is necessary since (1) trackers may falsely claim the HTML size attributes, and (2) the real trackers may hide behind the redirection.

Email Tracking Notification. For the sake of transparency, it is necessary to inform users when tracking is detected. Today, many websites are required (*e.g.,* by EU Privacy Directive) to display a notice to inform users when cookies are used for web tracking. More recently, EU's new GDPR policy forbids online services from tracking users with emails without unambiguous consent. However, there is no such privacy policy in the U.S.. While legislation may take a long time, a more immediate solution is to rely on email services or email clients to notify users.

A Comparison with Previous Research. The most related work to ours is a recent study that analyzed emails tracking of 902 websites (12,618 emails) [17]. In this work, we collect a dataset that is larger by orders of magnitude. Some of our results confirm the observations of the small-scale study. For example, we show that obfuscation is widely used to encode user identifiers for tracking and MD5 is the most commonly used method, both of which are consistent with [17]. Interestingly, Some of our results are different, in particular, the top third-party trackers (Table IX). For example, `doubleclick.net`, which was ranked 1_{st} by [17], is only ranked 7_{th} based on unique sender domains (ranked 2_{nd} based on email volume) in our dataset. `list-manage.com` was ranked 10_{th} by [17] but came to the top in our analysis. There are a couple reasons that may contribute to the differences. First, the previous work collected a small email dataset from 902 sender domains, while we collected emails from 210,000+ sender domains. Second, the previous study collected data from "Shopping" and "News" categories, while our dataset covers more than 100 website categories. Third, previous work only considered tracking URLs that contain an explicit user identifier (*i.e.*, email address), while we cover more tracking methods (*e.g.*, invisible or remote pixels).

VIII. Limitations

The first limitation is that our analysis only covers disposable email services with user-specified addresses (UA). This is mainly due to the difficulty to obtain data from randomly-assigned addresses (RA). Here, we use the small dataset collected from RA services (§III-A) to provide some contexts. Recall the dataset contains 1,431 messages from 5 RA services. After removing personal and non-English emails, we apply our classifier to the rest 1142 emails. We find that randomly-assigned addresses also contain account management emails, including 134 registration emails (11.7%), 44 password reset emails (3.9%), and 32 authentication emails (2.8%). We also notice that the spam email ratio is lower in RA services (81.6%) than that of UA services (94%). Intuitively, spammers often blindly send spam emails to addresses with popular usernames.

The second limitation is that our dataset is not representative with respect to a normal user inbox. Our measurement results cannot be used to assess email tracking at a per-user level. Instead, the main advantage of the dataset is that it contains emails sent by a large number of online services (including the top-ranked websites). This allows us to analyze email tracking from the perspective of online services (200K domains across 121 categories). For future work, we can evaluate the user-level tracking through user studies.

Third, for ethical considerations, we decided not to manually analyze the PII or back-track the accounts registered with the disposable addresses. This has limited our ability to answer some of the questions. For example, in §IV-A, we did not manually confirm the validity of detected PII, assuming the training accuracy transfers well to the testing. In §IV-C, it is possible that spammers would use the email addresses to register fake accounts in online services, but we cannot confirm. Similarly, for the password reset emails, it is possible that the emails were triggered by malicious parties who were trying to login other people's accounts, or by the real owners of the accounts who forgot the password.

Fourth, our email tracking detection is still incomplete. Theoretically, it is possible for a tracker to use subdomain names (instead of URL parameters) to identify individual users, or use font links (instead of image links). However, we did not find such cases in our dataset. In addition, our current method cannot detect tracking URLs that use both large tracking images and random strings as user identifiers.

IX. Related Work

Web Tracking and Email Tracking. Web tracking has been extensively studied by researchers in the past decade [15]. Researchers have analyzed third-party web tracking across different websites [29] and countries [23]. Consistently, different studies have shown that Google is the top tracker on the web [34], [43] where 80% of Alexa top 1 million websites have Google-owned trackers [31]. Web tracking has turned into a cat-and-mouse game. Researchers have studies various tracking techniques such as flash cookies [46], [12], canvas fingerprinting, evercookies, and cookie syncing [9], [18]. While adblockers help to reduce tracking, anti-adblockers are also increasingly sophisticated [59], [24], [36], [39].

Disposable Accounts and Phone Verified Accounts. Previous work has studied disposable SMS services where public phone numbers are offered to users for a temporary usage [41]. Researchers also studied the security risks of man-in-the-middle attack [20], and use the collected messages to investigate SMS spam [25], [37]. A recent work shows that "retired" addresses from popular email services can be re-registered to hijack existing accounts [21]. Other researchers looked in how disposable SMS are used to create phone-verified fake accounts in online services [50].

PII Leakage and Email Hijacking. Previous works have examined PII leakage under various channels [26], [27] such as mobile network traffic [42], [53], website contact forms [47], and cross-device tracking [14]. Our work differs from previous works with a focus on PII leakage during email tracking.

X. Conclusion

In this paper, we perform a first measurement study on disposable email services. We collect a large dataset from 7 popular disposable email services (2.3 million emails sent by 210K domains), and provide new understandings of what disposable email services are used for and the potential risks of usage. In addition, we use the collected email dataset to empirically analyze email tracking activities. Our results provide new insights into the prevalence of tracking at different online services and the evasive tracking methods used of trackers. The results are valuable for developing more effective anti-tracking tools for email systems.

Acknowledgment

We would like to thank our shepherd Manos Antonakakis and the anonymous reviewers for their helpful feedback. This project was supported in part by NSF grants CNS-1750101 and CNS-1717028, and Google Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

References

[1] Air force mypers. http://www.afpc.af.mil/Support/myPers/.

[2] Alexa top 1 million websites. https://www.alexa.com/topsites.

[3] Guerrillamail. https://www.guerrillamail.com/.

[4] Mailchimp. https://mailchimp.com/help/limitations-of-html-email/.

[5] Maildrop.cc privacy policy. https://maildrop.cc/privacy.

[6] Mailinator privacy policy. https://www.mailinator.com/faq.jsp.

[7] Selenium. http://www.seleniumhq.org/.

[8] Enron email dataset. https://www.cs.cmu.edu/~enron/, May 2015.

[9] Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., and Diaz, C. The web never forgets: Persistent tracking mechanisms in the wild. In *Proc. of CCS'14* (2014).

[10] Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., and Preneel, B. Fpdetective: dusting the web for fingerprinters. In *Proc. of CCS'13* (2013).

[11] Arshad, Sajjad, K. A. R. W. Include me out: In-browser detection of malicious third-party content inclusions. In *Proc. of Financial Cryptography and Data Security'17* (2017).

[12] Ayenson, M. D., Wambach, D. J., Soltani, A., Good, N., and Hoofnagle, C. J. Flash cookies and privacy ii: Now with html5 and etag respawning. In *SSRN* (2011). http://dx.doi.org/10.2139/ssrn.1898390.

[13] Ball, G. H., and Hall, D. J. Isodata, a novel method of data analysis and pattern classification. Tech. rep., Stanford research inst Menlo Park CA, 1965.

[14] Brookman, J., Rouge, P., Alva, A., and Yeung, C. Cross-device tracking: Measurement and disclosures. *Proc. of PETs'17* (2017).

[15] Budak, C., Goel, S., Rao, J., and Zervas, G. Understanding emerging threats to online advertising. In *Proc. of EC'16* (2016).

[16] Das, A., Bonneau, J., Caesar, M., Borisov, N., and Wang, X. The tangled web of password reuse. In *Proc. of NDSS'14* (2014).

[17] Englehardt, S., Han, J., and Narayanan, A. I never signed up for this! privacy implications of email tracking. In *Proc. of PETS'18* (2018).

[18] Englehardt, S., and Narayanan, A. Online tracking: A 1-million-site measurement and analysis. In *Proc. of CCS'16* (2016).

[19] Fifield, D., and Egelman, S. Fingerprinting web users through font metrics. In *Proc. of Financial Cryptography and Data Security'15* (2015).

[20] Gelernter, N., Kalma, S., Magnezi, B., and Porcilan, H. The password reset mitm attack. In *Proc. of IEEE S&P'17* (2017).

[21] Gruss, D., Schwarz, M., Wübbeling, M., Guggi, S., Malderle, T., More, S., and Lipp, M. Use-after-freemail: Generalizing the use-after-free problem and applying it to email services. In *Proc. of Asia CCS'18* (2018).

[22] Hosie, R. Ashley madison hacking: What happened when married man was exposed? Independent, 2017.

[23] Iordanou, C., Smaragdakis, G., Poese, I., and Laoutaris, N. Tracing cross border web tracking. In *Proc. of IMC'18* (2018).

[24] Iqbal, U., Shafiq, Z., and Qian, Z. The ad wars: retrospective measurement and analysis of anti-adblock filter lists. In *Proc. of the IMC'17* (2017).

[25] Jiang, N., Jin, Y., Skudlark, A., and Zhang, Z.-L. Greystar: Fast and accurate detection of sms spam numbers in large cellular networks using gray phone space. In *Proc. of USENIX Security'13* (2013).

[26] Krishnamurthy, B., Naryshkin, K., and Wills, C. Privacy leakage vs. protection measures: the growing disconnect. In *Proc. of the Web'11* (2011).

[27] Krishnamurthy, B., and Wills, C. E. On the leakage of personally identifiable information via online social networks. In *Proc. of the ACM workshop on Online social networks'09* (2009).

[28] Laperdrix, P., Rudametkin, W., and Baudry, B. Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In *Proc. of IEEE S&P'16* (2016).

[29] Lerner, A., Simpson, A. K., Kohno, T., and Roesner, F. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *Proc. of USENIX Security'16* (2016).

[30] Li, Y., Wang, H., and Sun, K. A study of personal information in human-chosen passwords and its security implications. In *Proc. of INFOCOM'16* (2016).

[31] Libert, T. Exposing the invisible web: An analysis of third-party http requests on 1 million websites. *International Journal of Communication* (2015).

[32] Luhn, H. Computer for verifying numbers, 1960. Patent No. 2,950,048.

[33] Ma, X., Hancock, J., and Naaman, M. Anonymity, intimacy and self-disclosure in social media. In *Proc. of CHI'16* (2016).

[34] Mayer, J. R., and Mitchell, J. C. Third-party web tracking: Policy and technology. In *Proc. of IEEE S&P'12* (2012).

[35] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS'13* (2013).

[36] Mughees, M. H., Qian, Z., and Shafiq, Z. Detecting anti adblockers in the wild. In *Proc. of PETs'17* (2017).

[37] Murynets, I., and Piqueras Jover, R. Crime scene investigation: Sms spam data analysis. In *Proc. of IMC'12* (2012).

[38] Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., and Vigna, G. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proc. of IEEE S&P'13* (2013).

[39] Nithyan, R., Khattak, S., Javed, M., Vallina-Rodriguez, N., Falahrastegar, M., Powles, J. E., Cristofaro, E., Haddadi, H., and Murdoch, S. J. Adblocking and counter blocking: A slice of the arms race. In *CoRR* (2016), USENIX.

[40] Piscitello, D. The new face of phishing. APWG, 2018.

[41] Reaves, B., Scaife, N., Tian, D., Blue, L., Traynor, P., and Butler, K. R. B. Sending out an sms: Characterizing the security of the sms ecosystem with public gateways. In *Proc. of IEEE S&P'16* (2016).

[42] Ren, J., Rao, A., Lindorfer, M., Legout, A., and Choffnes, D. Recon: Revealing and controlling pii leaks in mobile network traffic. In *Proc. of the MobiSys'16* (2016).

[43] Roesner, F., Kohno, T., and Wetherall, D. Detecting and defending against third-party tracking on the web. In *Proc. of NSDI'12* (2012).

[44] Rose, S., Engel, D., Cramer, N., and Cowley, W. Automatic keyword extraction from individual documents. In *Text Mining: Applications and Theory*. 2010, pp. 1 – 20.

[45] Seetharaman, D., and Bindley, K. Facebook controversy: What to know about cambridge analytica and your data. *The Wall Street Journal* (2018).

[46] Soltani, A., Canty, S., Mayo, Q., Thomas, L., and Hoofnagle, C. J. Flash cookies and privacy. In *AAAI spring symposium: intelligent information privacy management* (2010).

[47] Starov, O., Gill, P., and Nikiforakis, N. Are you sure you want to contact us? quantifying the leakage of pii via website contact forms. *Proc. of PETs'16* (2016).

[48] Starov, O., and Nikiforakis, N. Extended tracking powers: Measuring the privacy diffusion enabled by browser extensions. In *Proc. of WWW'17* (2017).

[49] Szurdi, J., and Christin, N. Email typosquatting. In *Proc. of IMC'17* (2017).

[50] Thomas, K., Iatskiv, D., Bursztein, E., Pietraszek, T., Grier, C., and McCoy, D. Dialing back abuse on phone verified accounts. In *Proc. of the CCS'14* (2014).

[51] Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., Markov, Y., Comanescu, O., Eranti, V., Moscicki, A., Margolis, D., Paxson, V., and Bursztein, E. Data breaches, phishing, or malware?: Understanding the risks of stolen credentials. In *Proc. of CCS'17* (2017).

[52] Ur, B., Segreti, S. M., Bauer, L., Christin, N., Cranor, L. F., Komanduri, S., Kurilova, D., Mazurek, M. L., Melicher, W., and Shay, R. Measuring real-world accuracies and biases in modeling password guessability. In *Proc. of USENIX Security'15* (2015).

[53] Vallina-Rodriguez, N., Kreibich, C., Allman, M., and Paxson, V. Lumen: Fine-grained visibility and control of mobile traffic in user-space.

[54] van der Nagel, E., and Frith, J. Anonymity, pseudonymity, and the agency of online identity: Examining the social practices of r/gonewild. *First Monday 20*, 3 (2015).

[55] Veras, R., Collins, C., and Thorpe, J. On semantic patterns of passwords and their security impact. In *Proc. of NDSS'14* (2014).

[56] Wang, C., Jan, S. T., Hu, H., Bossart, D., and Wang, G. The next domino to fall: Empirical analysis of user passwords across online services. In *Proc. of CODASPY'18* (2018).

[57] Wang, D., Zhang, Z., Wang, P., Yan, J., and Huang, X. Targeted online password guessing: An underestimated threat. In *Proc. of CCS'16* (2016).

[58] Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng, H., and Zhao, B. Y. You are how you click: Clickstream analysis for sybil detection. In *Proc. of USENIX Security'13* (2013).

[59] Zhu, S., Hu, X., Qian, Z., Shafiq, Z., and Yin, H. Measuring and disrupting anti-adblockers using differential execution analysis. In *Proc. of NDSS'18* (2018).

Appendix – Obfuscated User Identifier

To detect obfuscated user identifiers (*i.e.* email addresses) in the tracking URLs, we have tested 31 different hash/encoding functions. If the link's parameters contain the "*obfuscated version*" of the receiver's email address, then the image is considered as a tracking pixel. As shown in Table XI, we apply 31 hash/encoding functions on the receiver email address to look for a match. We also test two-layer obfuscations by exhaustively applying two-function combinations, *e.g.*, `MD5(SHA1())`. In total, we examine 992 obfuscated strings for each address.

TABLE XI: Functions to obfuscate user identifiers.

Hash or encoding functions (31 in total)
MD2, MD4, MD5, RIPEMD, SHA1, SHA224, SHA256, SHA384, SHA512, SHA3_224, SHA3_256, SHA3_384, SHA3_512, blake2b, blake2s, crc32, adler32, murmurhash 3 32 bit, murmurhash 3 64 bit, murmurhash 3 128 bit, whirlpool, b16 encoding, b32 encoding, b64 encoding, b85 encoding, url encoding, gzip, zlib, bz2, yenc, entity