Pervasive surveillance of the Internet presents two classes of risk related to information disclosure: an increase in the risk that individual flows' content may be revealed to the surveillor and a risk that cross-correlation of flows may reveal information to a surveillor that individual flows might not. This document primarily addresses the second set of risks. The amount of data available to the surveillor within a single flow impacts the ease of correlation, however, so they cannot be fully disentangled. If a flow reveals application-level tokens or data subject to fingerprinting, flows containing the same token or fingerprinting data may be correlated more easily. Rather than discussing this in detail, this paper describes only two states for individual flows: "open" and "confidential". These may be understood as endpoints on a spectrum ranging from all data within the flow being cleartext to all data other than destination being encrypted with a method resistant to feasible attacks.

Correlation of flows associated with an individual or endpoint presents an opportunity for a surveillor to build up an understanding of context. This context may, in turn, represent a risk for information leakage beyond that of the individual flows. Metadata analysis of surveillance records which showed a confidential flow from an endpoint to a health related website (e.g. a TLS-protected web search on WebMD) would obviously imply some loss of confidentiality, but this loss would be magnified if the initial flow could be correlated with a second flow to an insurance site and a third to a hospice care site. Even if all of these flows are confidential, significant information has passed to the surveillor. Further associations across time with phone calls or similar records can quickly build up further data on a context that the individual initiating these flows likely considers deeply personal.

Since much of the metadata analysis is possible based simply on the presence of the flows between a specific source and destination, this may appear to be intractable. There are, however, three classes of mitigation available which may increases the costs of surveillance past the threshold of specific pervasive surveillance actors. These are: aggregation, contraflow, and multipath.

Flow aggregation occurs when a set of endpoints shares an upstream service which combines their traffic; a common example is an HTTP proxy acting on behalf of an enterprise, educational institution, or other body. To get full access to information on the source of a flow, a surveillance point must be established prior to the aggregation point; this increases the costs of surveillance, potentially significantly. For this to work as a mitigation, however, it is important that the aggregating service minimize the amount of information traceable to the originating endpoint when it instantiates its flows to onward destinations. Cleartext requests sent through an HTTP proxy, for example, benefit from this mitigation only if the proxy eliminates data subject to fingerprinting prior to making the request to the origin server. Confidential requests benefit at all times.

It's also useful to note that the mitigation benefits of flow aggregation occur with confidential requests even when aggregation occurs at a single site which serves multiple types of content. A connection to a blog aggregation site, for example, does not necessarily reveal which blogs were read, provided that the flow target does not change with each blog. That means that a blog aggregation site blog reached via a URL like https://aggregation.example/blogname leaks less information than one reached via a URL like https://blogname.aggregation.example , since the DNS query for the hostname in the second provides data for the surveillance analysis to use in associating the confidential flow.

Contraflow occurs when a node sends traffic via a path which does not follow the IP routing for that flow, commonly using a tunnel or set of tunnels. This creates an increased cost for a potential surveillor by requiring the correlation to encompass a larger set of paths to detect individual flows; it may also increase costs by requiring cooperation among a larger set of state actors. For a tunnel to be effective at providing an effective mitigation as contraflow, it must hide the original source and destination; GRE and similar methods are too simple for a surveillance analysis to unwrap to provide an effective mitigation. Onion routing, as exemplified by TOR, is the best known contraflow method built to defeat surveillance; uProxy builds on WebRTC data channels to create a similar contraflow effect. Beyond those purpose-built methods, some forms of VPNs, including those built to evade country-based content restrictions, provide similar mitigation. Note that one common characteristic of contraflow is that it increases the number of route miles used to carry the flow. That has implications both for overall latency and for the capacity required by the networks serving these flows; where contraflow crosses paid peering or transit links, it may also imply new costs to the serving networks.

Multipath occurs when some flows are associated with different links than others, even though the flows originate at the same endpoint. A common form of multipath occurs when a host has a split-tunnel VPN, which uses the tunnel for some flows but the local link for others. Another common form occurs when a mobile device moves between 802.11 and cellular uplinks. Multipath increases the costs of surveillance by forcing the analysis to encompass a larger set of flows and to divorce what may be "trigger" interactions from the flows with which they are associated. If, for example, a surveillance analysis relies on DNS to trigger further investigation, the common split-tunnel configuration presents a problem: DNS queries passed through the tunnel may cause flows which are then routed via the local link. (In the example above, "https://blogname.aggregation.example/" and all the other virtual hosts under aggregation.example might be served by the same set of IPs; the DNS query would be required to provide the signal that a blog of interest was accessed.) Unless there are other available tokens or data which fingerprints the flow initiator, the relationship between the two is lost; it is more difficult to correlate in any case.

In the above taxonomy, the mitigations are described as "occurring" rather than being initiated because in most cases the mitigation is a side-effect of some other desired effect (ToR and uProxy being exceptions). The question we now raise is how to deploy these, either singly or in combination, to provide practical means to generally raise the cost of surveillance without unacceptable impact to the network latency, throughput, or loss.

The easiest deployment of aggregation is actually in the content server domain, where the ability to access content at URLs which do not leak information is relatively easy--path and query elements in the URI being generally sufficient for disambiguation. Outside of this, aggregation can be done for some protocols (HTTP, SIP, email) with elements already specified and widely available. The value of these varies widely, however, in part based on the user population covered and in part based on other protocol interactions (such as the later set-up of peer-to-peer flows in SIP). A proxy with a very large user population will likely become a target for infiltration; one too small may have insufficient aggregation to avoid fingerprinting. One possible configuration to avoid both risks is to deploy a number of small proxies which would be infeasible to infiltrate but which use proxy chaining to send their traffic via other proxies. By varying the path through the proxies, any proxy can both ensure that the user population it serves is being further aggregated with other proxies’ and ensure that no long-term association creates a hierarchical system. An enterprise with multiple offices might, for example, deploy a proxy in each but have traffic variably routed among the proxy set before being sent onward. As noted above, this does require the proxies to remove data suitable for fingerprinting; it also requires the final proxy to remove any internal routing information (such as the HTTP Via header or the Received header) for non-confidential channels. The set of available proxies must also not increase latency beyond that acceptable to the end users.

The method described above crosses into contraflow, so it might be considered a hybrid approach. The VPN mechanisms currently deployed to handle geography-specific content can similarly utilize proxies to aggregate requests for content and some of the commercial entities (e.g. http://www.ukproxyserver.com/) offer this service. One advantage of the combination is that an aggregating proxy is likely to be protocol or application specific, meaning that someone wishing to use it for some but not all purposes can do so easily (one browser using a proxy in territory X and another in territory Y).   This creates the opportunity to combine contraflow with a relatively simple form of multipath. A side effect of these being commercial services is that those sharing the service are simply jointly customers of the service, and have no other necessary relationship, where an organization-based proxy or a social peer-to-peer system provides other relationship signals to the surveillor.

The use of any contraflow technique necessarily increases latency, and its consistent use may eventually defeat the point; it may simply shift who is conducting the surveillance, rather than eliminating the threat. That may be sufficient for some privacy purposes, but simply making its use occasional allows it to add a simple form of multipath, especially if that usage is not triggered by specific privacy concerns. Setting up and using such tunnels both when specific privacy concerns are in play and at random other times may provide better masking.

The dedicated multipath systems like ToR and uProxy combine the features of aggregation and contraflow; exit nodes carry flows from multiple users and emit traffic from network elements unrelated to the source. Larger scale use of them and of similar systems would be clearly beneficial, both in providing direct privacy benefits and reducing the signal related to their use. As noted above, though, methods that provide some of the same benefits and relate to commercial VPN services may be equally good. Whether traffic is steered onto the VPN based on type, time, or some more random function, their use increases costs to the surveillor at relatively light configuration costs and with relatively good masking. As ever, latency and reliability may suffer, and some variability in content served may be evident.

In summary, pervasive encryption is the most effective mitigation to pervasive surveillance, but it must be accompanied by methods that limit metadata and correlation analysis to be truly effective. The use of aggregation, contraflow, and multipath can each contribute to increasing the cost of that analysis.   Protocol and system designers are advised to ensure that their systems work well in the presence of these mitigations. To be most effective, these mitigations must become sufficiently pervasive that their use is not a practical signal for targeted surveillance. That likely means combining them into systems which allow large number users to share the facilities provided; this can be done in enterprises, cooperating organizations, commercial endeavors, or peer-to-peer networks.