Network Management of Encrypted Traffic

Version 1.0

28 February 2015

This is a Position Paper of the GSMA

Security Classification: Non-confidential

Copyright Notice

Disclaimer

The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Antitrust Notice

The information contain herein is in full compliance with the GSM Association’s antitrust compliance policy.

Table of Contents

1 Introduction

1.1 Overview

Encrypted Web traffic is expected to continue its upward trend, driven by increased privacy awareness, uptake by major players and advocacy from the IETF and W3C. This document describes the technical details of options to persist certain network management functions for encrypted traffic. These options are categorised into two main sections: first, where traffic has been encrypted by opt-in to a browser proxy, and second, where HTTPS is used end-to-end between the content server and client . There are existing and future solutions described in the document in addition to advice on whether the mitigation option requires single or joint operator action to proceed.

1.2 Scope

This document is aimed towards technical architects with knowledge of the operator network traffic management functions, browser vendors and middleware/proxy vendors. Information on non-network impacting workarounds is also provided.

The document captures the mitigations identified to ensure that operators can execute on their traffic management requirements, as identified in the proxy use case document, in IETF HTTP BIS Working Group [Ref 1] . These use cases have been broadly categorised, and each mitigation will indicate the category of use case that it applies to. Where a mitigation option involves a technical integration at the network, that information is also detailed.

1.3 Abbreviations and Definition of Terms

Term	Description
Accommodating or Browser Proxies	Proxies that optimise web traffic for mobile browsers. These proxies cache, compress and accelerate traffic delivery. Opera Mini is the most common example.
Browser	A web browser; most commonly: Google Chrome, Mozilla's Firefox, Apple's Safari, Opera, or Microsoft's Internet Explorer.
HTTP/1.1	Existing version of the HTTP protocol most often used.
HTTP/2	The evolution of the HTTP protocol. Based on Google's SPDY protocol and currently being designed by the IETF.
HTTP:// or HTTPS:// URIs	URIs (or URLs, in this context they are the same thing) which start with HTTP:// or HTTPS://. HTTPS:// URIs are addressed which are being carried over TLS or SSL.
TLS for HTTP:// URIs, HTTPS:// URIs	TLS is defined below. These two terms refer to the use of TLS for HTTP:// URIs where a content provider has not enacted TLS encryption themselves but the browser vendor has done so on their behalf. TLS encryption for HTTPS:// URIs is where a content provider has made the conscious effort to encrypt their content by purchasing a certificate, configuring their servers and serving resources using HTTPS:// URIs.
IETF	Internet Engineering Task Force. An individual-membership based organisation which set standards for the internet.
Opportunistic Encryption	When a browser chooses to upgrade HTTP/2 traffic on an HTTP:// URI to utilise TLS using a 'relaxed' version of TLS. Please see the References section for the IETF draft which explains this in detail.

Term	Description
QoS	Quality of Service.
SPDY	Pronounced ‘speedy’. Google’s originally designed new version of the HTTP/1.1 protocol. Due to its maturity it was used as the basis for HTTP/2.0, but it is not the same as HTTP/2.
TCP	Transmission Control Protocol. A Transport Layer protocol which carries much of the internet’s traffic. TCP operator’s bandwidth control and in-order packet delivery making it a ‘reliable’ protocol.
TLS	Transport Layer Security, a cryptographic protocol which is designed to provide communication security over the Internet. TLS encrypts data at the Application Layer and offers end-to-end encryption by the use of certificates and key exchange.
Trusted Proxy	A proxy that runs separate TLS sessions between the client and the proxy, and the proxy and the content server to allow operator and other services to still function for TLS based traffic. A user will have the power to ‘opt-in’ or ‘opt-out’ of using this proxy.
User Agent	A web browser (in this context).
Website, Webpage, Webapp	Synonymous terms for web content delivered by HTTP (in most cases, video content may be delivered by other protocols).

1.4 Document Cross-References

Ref	Document Number / Name	Title
[1]	IETF Proxy Use Case Stories	HTTPS://github.com/HTTP2/HTTP2-spec/wiki/Proxy-User-Stories
[2]	Google DCP	Google Data Compression Proxy
[3]	Google PBF	Google Proxy Bypass Filters
[4]	IETF TLS Extension	HTTP://tools.ietf.org/html/rfc6066#page-6
[5]	IETF Server Name Indication Extension	IETF Server Name Indication Extension
[6]	IETF DiffServ	IETF RFC 2474 - Defintion of Differentiated Services Filed
[7]	IETF ECN	IETF RFC 3168 - Explicit Congestion Notification to IP
[8]	IETF MPLS	IETF RFC 3031 – Multiprotocol Label Switching Architecture
[9]	IETF DiffServ	IETF RFC 2475 – Architecture for Differentiated Services
[10]	IETF EAP	IETF – Explicitely Auhtenticated Proxy draft
[11]	IETF HTTP2 draft	HTTP://tools.ietf.org/html/draft-ietf-HTTPbis-HTTP2-14.
[12]	IETF HTTP2 implementation	HTTPS://github.com/HTTP2/HTTP2-spec/wiki/Implementations
[13]	IETF JSON Web Encryption	HTTP://tools.ietf.org/html/draft-ietf-jose-json-web-encryption-31
[14]	W3C Sub Resource Integrity	HTTP://www.w3.org/TR/SRI
[15]	Apple iOS Understanding restrictions	HTTP://support.apple.com/kb/HT4213

[16]	Android Net Nanny app	HTTP://www.i-technosoft.co.uk/android-software/netnanny-android
[17]	AVG' Family Safety solutions	iOS: HTTPS://itunes.apple.com/us/app/avg-family-safety/id520773859?ls=1&mt=8 Windows Phone: HTTP://www.windowsphone.com/en-gb/store/app/avg-family-safety-8/1df94db9-d1b4-4a6a-942e-4e04c97fb32c
[18]	IETF TCPINC	HTTP://tools.ietf.org/html/draft-bittau-tcpinc HTTP://tools.ietf.org/html/draft-rescorla-tcpinc-tls-option HTTP://tools.ietf.org/html/draft-thomson-tcpinc-dtls HTTP://tools.ietf.org/html/draft-touch-tcpm-tcp-edo
[19]	IETF TCPCrypt description	HTTP://tools.ietf.org/id/draft-bittau-tcpinc-01.txt
[20]	IETF TCPINC TLS options	HTTP://tools.ietf.org/id/draft-rescorla-tcpinc-tls-option-00.txt
[21]	IETF TLSINC DTLS options	HTTP://tools.ietf.org/id/draft-thomson-tcpinc-dtls-00.txt
[22]	IETF Extended TCP Authenticaiton	HTTP://tools.ietf.org/html/draft-touch-tcpm-tcp-edo
[23]	IETF Denial of Service Risk	HTTP://tools.ietf.org/html/rfc6437#section-6.2
[24]	IETF TLS 1.3 description	HTTP://tools.ietf.org/html/draft-ietf-tls-tls13-02
[25]	IETF Opportunistic Encryption	HTTPS://tools.ietf.org/html/draft-ietf-HTTPbis-HTTP2-encryption-00
[26]	IEFT Alternate Services	HTTPS://tools.ietf.org/html/draft-ietf-HTTPbis-alt-svc-03
[27]	Increasing HTTP Transport Confidentiality with TLS Based Alternate Services (P. McManus – Mozilla)	HTTPS://www.w3.org/2014/strint/papers/20.pdf
[28]	Google QUIC protocol explained	HTTPS://docs.google.com/document/d/1lmL9EF6qKrk7gbazY8bIdvq3Pno2Xj_l_YShP40GLQE
[29]	QUIC Crypto documentation	HTTPS://docs.google.com/document/d/1g5nIXAIkN_Y-7XJW5K45IblHd_L2f5LTaDUDwvZ5L6g
[30]	TLS with Perfect Forward Secrecy	HTTP://security.stackexchange.com/questions/7690/what-should-i-know-before-configuring-perfect-forward-secrecy
[31]	IEFT TCPINC Research project documentation references	1. HTTP://tools.ietf.org/wg/tcpinc/charters 2. HTTP://tools.ietf.org/html/draft-bittau-tcpinc 3. HTTP://tools.ietf.org/html/draft-rescorla-tcpinc-tls-option 4. HTTP://tools.ietf.org/html/draft-thomson-tcpinc-dtls 5. HTTP://tools.ietf.org/html/draft-touch-tcpm-tcp-edo
[32]	Attribute Based Encryption and Routing	HTTP://conferences.sigcomm.org/sigcomm/2013/papers/sigcomm/p513.pdf
[33]	TCP Minion explained	HTTP://dedis.cs.yale.edu/2009/tng/papers/nsdi12.pdf

[34]	IETF SPUD (Protocol Inside UDP)	HTTP://www.ietf.org/proceedings/90/slides/slides-90-appsawg-7.pdf
[35]	IETF Multipath TCP (RFC 6824)	HTTPS://tools.ietf.org/html/rfc6824
[36]	IETF CDN Interconnect	HTTPS://datatracker.ietf.org/wg/cdni/charter/

2 Recommendations for Encrypted Mitigation options

'Mitigation' is used throughout this document, and refers to a method of supporting a particular use case when traffic is encrypted. Encryption is positive from an end user privacy perspective and it should be applied where possible, however it has an impact on existing traffic management practices.

A 'mitigation' does not simply imply network-based software or policy enforcement; it may also include device-based alternatives or external industry efforts that support both encryption and certain traffic management requirements. Only mitigations involving a technical change are covered here (although that technology can be applied outside the network in some cases).

HTTPS means a secure, certificate-based connection between the client and origin server (i.e. the origin of the content being requested). This is often referred to as 'end-to-end' encryption. The request made by the client is an 'HTTPS' scheme URI.

HTTP over Transport Layer Security (TLS) is used in this document to refer to an 'HTTP' URI scheme request that has been actively upgraded to travel over a secure connection. Typically this refers to an 'HTTP' request made to a browser proxy, which has created an encrypted connection to the client.

In this document, non-mobile networks are out of scope, however a future version may consider fixed networks (whether directly connected or via a Wi-Fi hub). Regulatory and non-technical mitigations are covered by a parallel GSMA activity.

1) The mitigations listed are not mutually exclusive. Therefore, the operator is encouraged to consider the full range of mitigations with a view to implementing a combination that best meets their regulatory and business requirements.
2) Existing mitigations can be applied today at a per-operator level.

Operators are particularly encouraged to undertake collective action and support for long term technical mitigations:

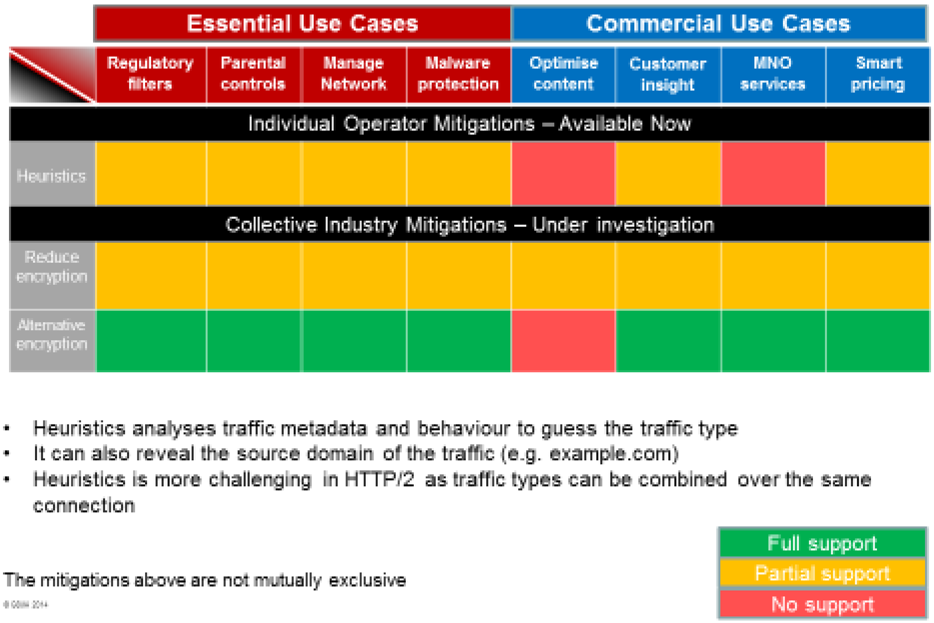
- Service Provider Proxy
- Uptake of HTTP/2 without transport layer encryption
- Object Level Encryption
- Further information on how to support these activities will be socialised with the ENCRY subgroup.

- The non-technical recommendations should also be considered.

2.1 Mitigations that are not recommended (out of scope)

1) Any mitigation which implements ‘Man in the Middle’ certificate spoofing
2) Any breaking of HTTPS/end-to-end encryption. End-to-end encryption here means encryption between a client and origin server, rather than encryption between a client and browser proxy.
3) Any ‘across the board’ blocking of a particular browser proxy

3 Summary of technical mitigations



	Essential Use Cases				Commercial Use Cases			
	Regulatory filters	Parental controls	Manage Network	Malware protection	Optimise content	Customer insight	MNO services	Smart pricing
Individual Operator Mitigations – Available Now								
Heuristics	🟨	🟨	🟨	🟨	🟥	🟨	🟥	🟨
Collective Industry Mitigations – Under investigation								
Reduce encryption	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
Alternative encryption	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩

- Heuristics analyses traffic metadata and behaviour to guess the traffic type
- It can also reveal the source domain of the traffic (e.g. example.com)
- Heuristics is more challenging in HTTP/2 as traffic types can be combined over the same connection

The mitigations above are not mutually exclusive

© GSMA 2014

🟩	Full support
🟨	Partial support
🟥	No support

Table 1: Technical mitigation options to HTTPS encryption

This table summarises which technical mitigations can be applied against which use cases. For the sake of brevity the mitigations have been grouped in the table:

1) ‘Heuristics’ refers to Heuristics and TLS Server Name Indication extension
2) ‘Google Toolkit’ refers to Google Canary URL, Google Proxy Bypass filters and Deploying Google Proxy within the MNO network
3) ‘Reduce encryption’ refers to Encouraging uptake of HTTP/2 without transport layer encryption

‘Alternative encryption’ refers to Object Level Encryption, and to some extent Device and cloud based filters.

4 Analysis of mitigations

4.1 Existing mitigations

4.1.1 Disabling encryption to the Google DCP: Google Canary URL

Reference documentation: Ref [2]

Description: When launching the browser on the mobile device or when switching from Wi-Fi to the cellular network, Google Chrome for Mobile issues a single unencrypted HTTP request to a well-known URL (HTTP://check.googlezip.net/connect). The mobile network can intercept this request and decide on the HTTP response to send to the browser according to the operator policy and the end user identity (see example below).

Note 'identity' can include a broader notion of geography or a group of users sharing some common trait such as a certain type of subscription (e.g., Facebook zero rating).

The intercepting component of the mobile network can either send a successful response to indicate to the browser to encrypt the connection, or the network can send any other HTTP error status code to inform the browser that encryption should not be applied to this particular user. This mechanism allows the network operator to enable or reject an encryption between a specific user and the Google DCP (Data Compression Proxy).

Scope: Google Chrome for Mobile users that have selected 'Reduce Data Usage' setting, and intended to allow encryption to be disabled on a per-user basis. Google do not document when the network may reject or enable encryption: an example usage is to disable encryption for users not proven to be adults. Note that Google Data Compression Proxy optimisations still occur even if the content is not encrypted.

Integration:

Two processes are required

1. Detect the request from the Chrome for Mobile browser.

The requested URL is HTTP://check.googlezip.net/connect.This will travel unencrypted through to the Packet Data Network Gateway (PDN-GW in LTE, or equivalent in other networks), therefore could be trapped at several nodes, however point (2) below will dictate the optimal node for a given network.

2. Determine any policy rules for the requesting user

This requires that the node performing (1) above has access to the requesting user's identity, and the ability to lookup policy rules set for that identity (or groups that the identity is a member of).

1) In practice, this means handling this at an enforcement point controlled by the PCRF (Policy Charging Rules Function) interface
2) The user identity may have been extracted from either an HTTP request header, or from an additional lookup (such as IMSI lookup based on allocated IP address of the terminal).

Should this process determine any policy rules applying to the requesting user, then the system will decide whether the rule(s) can allow or disallow encryption to occur.

Example:

Example_Operator categorises all customers as either ‘child’ (the default) or ‘adult’. The customer must present credentials to the retail store to achieve ‘adult’ categorisation.

User *Ch* is a ‘child’ and user *Ad* is an ‘adult’ connected to the Example_Operator network. Both use Google Chrome for Mobile, and have set ‘Reduce Data Usage’ in their settings. When they start the browser, process (1) and (2) described above occur, with the result:

- User *Ch* receives the HTTP response ‘403 Forbidden’ from the Example_Operator network. This means that the Chrome for Mobile traffic may still utilise the Google DCP but encryption will not occur. Hence Example_Operator has visibility of the HTTP traffic en route to the Google DCP, and content filters for ‘adult content’ can be applied.

- User *Ad* receives the response HTTP ‘200 OK’ from the Example_Operator network. Hence Chrome for Mobile initiates an encrypted tunnel to the Google DCP, and Example_Operator network will not filter for adult content delivered over this tunnel.

Notes:

It is not clear whether a blanket rule of ‘no encryption’ for all users is supported by Google (in other words, whether the Canary URL test can be ‘failed’ in all cases).

Mitigates the use cases:

“Regulatory filtering”, “parental control”, “customer malware protection” when applied to child versus adult SIMs, but only for traffic proxied by the Google DCP i.e. HTTP over TLS requests made by Chrome for Mobile browsers with the DCP activated.

Limitations:

The Google Canary URL solution is specific for the Google DCP. The corresponding solution in the operator network will not work for browser proxies of other companies. Moreover, there is no standard in place which would define the required behaviour of the browser. Thus, proprietary changes to the browser can lead to the solution implemented by the operator not working anymore.

The number of users of Google Chrome for Mobile who have selected ‘Reduce Data Usage’ (and hence activated the Google DCP) is understood to be low. Therefore the ability to apply the Canary URL consistently across a high number of users is untested.

Relative cost to implement: low

4.1.2 Google Proxy Bypass filters

Reference documentation: Ref [3]

Description: a blacklist hosted at the Google DCP. Populated by Google based on operator submissions.

Scope: Google Chrome for Mobile users that have selected ‘Reduce Data Usage’ setting. Differs from the Canary URL process in that it is intended to be applied across all users, rather than a specific user or group.

Integration: The operator is required to submit a list of URLs to Google that they wish to bypass the DCP. This submission is an offline process; operators should submit URLs to their Google contact for attention of the DCP team to implement. Once the bypass URLs have been integrated at Google, then the following behaviour is expected:

1. The Google DCP will detect any request to this URL
2. Google will suspend encryption for a period of 1 to 5 minutes between the requesting Chrome for Mobile browser and the DCP
3. The DCP will respond to the Chrome for Mobile browser and make it reissue the request unencrypted.
4. This will allow operator filters to detect the requests made during the 1-5 minute suspension window and act accordingly.

Example: Example_Operator provides several service portals for their mobile customers. These are hosted within their network (rather than the public Internet) in order to provide seamless authentication and lookup of sensitive customer data. If the DCP is being utilised, then Google’s DNS server will be used to resolve the request. Therefore, the portal URIs/IP addresses must be submitted to the Google Proxy Bypass list, since a Google DNS will not be able to resolve them to an IP address. Any request to the Example_operator service portal will therefore result in steps 1-4 in the ‘Integration’ section above being followed, and the DCP will not proxy the request.

Notes: Note that by default Google bypass the proxy for international blacklists, namely the Internet Watch Foundation list (child abuse materials). Note, ‘bypass’ means the steps 1-4 above in the ‘Integration’ section are performed. The documentation mentions that national court order lists are also included, however gaps have been identified in these lists, possibly because Google may not be party to blocking issued by national telecoms regulators. Therefore, operators are encouraged to determine the result of Google DCP usage when such court order blocked URIs are requested (for example, offshore gambling sites).

Mitigates the use cases:

“Regulatory filtering”, “MNO Services” *but only for traffic proxied by the Google DCP. I.e. HTTP over TLS requests made by Chrome for Mobile browsers with the DCP activated.*

Limitations:

This solution works only together with the Chrome for Mobile browser/Google Data Compression Proxy.

Despite the name, the DCP is not bypassed. The solution is not ideal for operator portals since the logic to go through the ‘bypass’ steps is stored at the DCP. This introduces latency as the request must be submitted via an encrypted tunnel to DCP, filtered, rejected, encryption torn down, and the request resubmitted in the clear.

The Proxy Bypass is intended for a limited number of sites; mainly operator billing portal and operator intranet sites. It can also be used to supplement Google’s understanding of country based restrictions on particular third party sites, for regulatory reasons. It is not intended to allow operators to bypass encryption for third party sites that they have commercial agreements with.

The URL submission process is a manual email to a Google contact, and therefore not real-time.

Relative cost to implement: low

4.1.3 Deploying Google Proxy within the operators' network

This is a per operator bilateral business decision to discuss with Google, and not covered by this document. Integration, relative cost and limitations are dependent on operator architecture, and any existing interface of Google infrastructure in their network (such as a Google cache).

4.1.4 TLS Server Name Indication extension

Description: Reference documentation: "Transport Layer Security (TLS) Extensions: Extension Definitions" Ref [4]

When initiating the TLS handshake, the Client may provide an extension field (server_name) which indicates the server to which it is attempting a secure connection.

TLS SNI was standardized in 2003 to enable servers to present the "correct TLS certificate" to clients in a deployment of multiple virtual servers hosted by the same server infrastructure and IP-address. Although this is an optional extension, it is today supported by all modern browsers, web servers and developer libraries. Notable exceptions are Android 2.2 and Internet Explorer 6 on Windows XP.

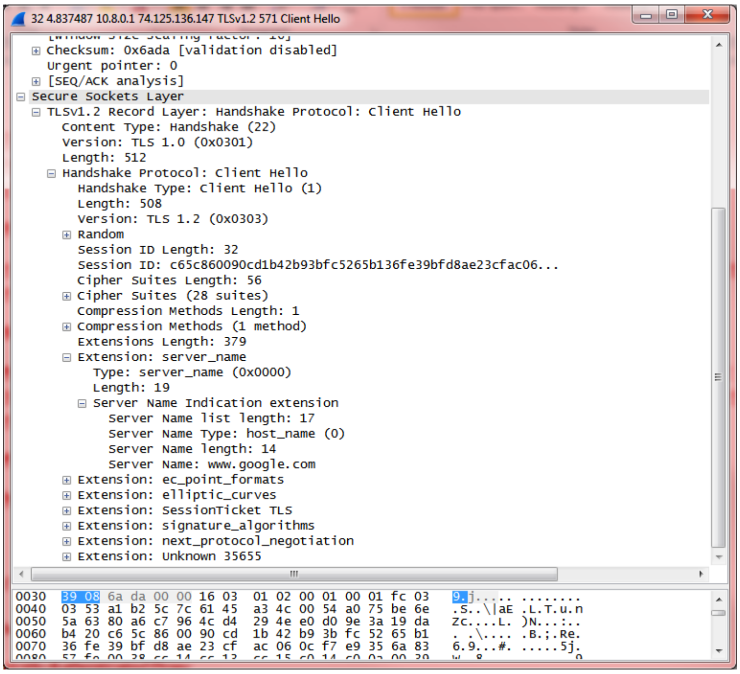
It should be noted that HTTP/2 introduces the Alt-SVC method for upgrading the connection from HTTP/1 to either unencrypted or encrypted HTTP/2. If the initial HTTP/1 request is unencrypted, the destination alternate service name can be identified before the communication is potentially upgraded to encrypted HTTP/2 transport.

HTTP/2 implementations MUST support the Server Name Indication (SNI) extension Ref [5]

Scope: All TLS connections that include a server_name extension.

Integration: Existing network filters will already be able to see the unencrypted TLS handshake. Therefore, parsing for the server_name extension of the TLS Handshake protocol 'Client Hello' will return the server name, if provided by the client.

Example:



```
32 4.837487 10.8.0.1 74.125.136.147 TLSv1.2 571 Client Hello
  ⊞ Checksum: 0x6ada [validation disabled]
    Urgent pointer: 0
  ⊞ [SEQ/ACK analysis]
⊟ Secure Sockets Layer
  ⊟ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    ⊟ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 508
        Version: TLS 1.2 (0x0303)
      ⊞ Random
        Session ID Length: 32
        Session ID: c65c860090cd1b42b93bfc5265b136fe39bfd8ae23cfac06...
        Cipher Suites Length: 56
      ⊞ Cipher Suites (28 suites)
        Compression Methods Length: 1
      ⊞ Compression Methods (1 method)
        Extensions Length: 379
      ⊟ Extension: server_name
          Type: server_name (0x0000)
          Length: 19
        ⊟ Server Name Indication extension
            Server Name list length: 17
            Server Name Type: host_name (0)
            Server Name length: 14
            Server Name: www.google.com
      ⊞ Extension: ec_point_formats
      ⊞ Extension: elliptic_curves
      ⊞ Extension: SessionTicket TLS
      ⊞ Extension: signature_algorithms
      ⊞ Extension: next_protocol_negotiation
      ⊞ Extension: Unknown 35655

0030  39 08 6a da 00 00 16 03  01 02 00 01 00 01 fc 03   9.j..... ........
0040  03 53 a1 b2 5c 7c 61 45  a3 4c 00 54 a0 75 be 6e   .S..\|aE .L.T.u.n
0050  5a 63 80 a6 c7 96 4c d4  29 4e e0 d0 9e 3a 19 da   Zc....L. )N...:..
0060  b4 20 c6 5c 86 00 90 cd  1b 42 b9 3b fc 52 65 b1   . .\.... .B.;.Re.
0070  36 fe 39 bf d8 ae 23 cf  ac 06 0c f7 e9 35 6a 83   6.9...#. .....5j.
```

Figure 1:Example of trace information for SNI

This packet capture shows a Server Name of www.google.com when a TLS Handshake is started with the Google DCP.

Limitations: This solution only works if the browser is populating the Server Name Indication extension. This need not be done, but may be done as per TLS standard. Therefore, even if existing network filters look out for seeing a Server Name Indication extension, they may not find one. The per-domain nature of SNI may not reveal the specific service or media type being accessed, especially where the domain is of a provider offering a range of email, video, Web pages etc. For example, certain Facebook or Twitter feeds may be deemed ‘adult content’, but the Server Name Indication will only indicate the server domain (facebook.com, twitter.com) rather than a URL path to be blocked.

Mitigates the use cases: “Regulatory filtering”, “parental control”, “customer QoE”, “MNO smart pricing” but only at a per-domain level. It can therefore be considered a coarse tool.

Relative cost to implement: low/medium

4.1.5 Heuristics

Description:

All IP packets are associated with meta data that can be utilized for identification and classification of traffic. The data available are: Source IP-address, destination IP-address, protocol (TCP, UDP), source port and destination port (collectively known as 5-tuple). In many scenarios, this makes it possible to directly classify the traffic related to a specific server/service even when the traffic is fully encrypted.

If the server/service is co-located on an infrastructure with other services that shares the same IP-address, the encrypted traffic cannot be directly classified. However, commercial traffic classifiers today typically apply heuristic methods, using traffic pattern matching algorithms to be able to identify the traffic. As an example, classifier products have for years been able to identify Skype using heuristic methods although the traffic is encrypted and mostly peer-to-peer.

Heuristics can be used to map given input data to particular conclusions via some heuristic reasoning.

Example of input data: IP destination address, TCP destination port, server name from SNI, typical traffic pattern (e.g. occurrence of IP packets and TCP segments over time).

Example of output data: A prediction for the name of the service or application being used (e.g. Skype video, Skype messaging).

Scope: Heuristics can be used to draw a conclusion or provide an estimate or answer a question like “Which application or service is being used by a consumer?”

Integration: Heuristics will be used by vendors of traffic inspection or management functions.

Mitigates the use cases: Customer QoE”, “MNO Customer insight”, ”MNO Network optimization” typically at a more granular level of detail than using TLS SNI alone.

Limitations: Heuristics are a way for vendors to differentiate their products. Therefore, products may differ substantially in terms of how ‘clever’ the heuristics algorithms are. Heuristics algorithms will have to be updated over time, as the shapes, patterns and characteristics of application traffic won’t remain constant over time. This can lead to ongoing costs for upgrades and maintenance of traffic inspection equipment or software; however that depends on implementation and commercial agreement.

Relative cost to implement: medium/high (variable according to complexity of heuristics offered)

4.1.6 Traffic management at lower network layers

Reference documentation:

- DiffServ Ref [6]
- Explicit Congestion Notification Ref [7]
- Multiprotocol Layer Switching (MPLS) Ref [8]

4.1.6.1 DiffServ:

Description: Data packets are flagged with a traffic class (class of service). Network operators may honour a DiffServ classification entering their network, or may choose to override it (since it is potentially open to abuse by a service provider that classifies all its content as high priority). The purpose is to help manage traffic and congestion in the network.

Integration: Please refer to Architecture for Differentiated Sevices (Ref [9]) or vendor-specific documentation.

Mitigates against: This partially mitigates against all cases where the IP address of the content service is known; however there are significant limitations.

Limitations: This requires the content provider to flag data packets. This is extra work for the provider, and it has potential for abuse if a content provider simply flags all packets with high priorities. The network would need to know which flags to trust and which to override.

The use of DiffServ within the operator network is beneficial where the operator determines the class of service itself; but where content is encrypted then heuristics would be needed to predict the traffic type entering the network.

HTTP/2 allows several streams to be multiplexed over a single TCP connection. This means that if a provider decides to send Web pages, videos, chat etc. as individual streams over the same connection, then DiffServ would be useless as it would apply to the TCP/IP connection as a whole. However it may be more efficient for such Web providers to serve each content type from separate, dedicated servers – this will become clearer as HTTP/2 deployments are tuned for optimal delivery.

Relative cost to implement: high (without existing DiffServ infrastructure), medium (with existing DiffServ infrastructure). Also requires content provider support.

4.1.6.2 Explicit Congestion Notification

Description: Explicit Congestion Notification (ECN) routers can exchange congestion notification headers to ECN compliant endpoints. This is in preference to inferring congestion from dropped packets (e.g. in TCP). The purpose is to help manage traffic and congestion in the network.

Integration: This solution required to be implemented at network and service provider. The service provider will utilise the ECN to reduce throughput until it is notified that congestion has eased.

Mitigates against: This partially mitigates against all cases where the IP address of the content service is known; however there are significant limitations.

Limitation: As with DiffServ, operators may not trust an external entity to mark packets in a fair/consistent manner.

Relative cost to implement: high (without existing ECN infrastructure), medium (with existing ECN infrastructure). Also requires content provider support.

4.1.6.3 Multi- Packet Label Switching (MPLS)

Description: on entering an MPLS-compliant network, IP packets are flagged with a ’Forward Equivalence Class’ (FEC). This allows the network to make packet-forwarding decisions according to their latency requirements. MPLS routers within the network parse and act upon the FEC value. The FEC is set according to the source IP address and port. The purpose is to help managing traffic and congestion in the network.

Integration: Likely to already exist in operator networks and national backbone networks. Requires deployment of an MPLS ‘backbone’ with label-aware switches/ routers. Complexity will depend upon the existing network topology.

Mitigates against: “Regulatory filtering”, “parental controls” (BUT only where the IP address of blocked sites is known in advance and hence such sites can be restricted at the IP layer)

Limitations: some entity has to maintain an up-to-date correspondence table between a website to block (parental controls and regulatory filtering use cases) and the IP address of a server. Then, this category of traffic has to be consistently mapped to a certain known set of MPLS labels which entails a great cost of setting it up and a recurring cost of maintaining it

Note: The difference between DiffServ and MPLS:

MPLS can specify how OSI Layer 3 (IP layer) traffic can be routed over Layer 2 (Data Link); DiffServ only operates over Layer 3. DiffServ is potentially a less complex integration as it is applied at the network edge servers only.

Relative cost to implement: high (without existing MPLS infrastructure), medium (with existing MPLS infrastructure). It also requires content providers' support.

4.2 Mitigations under investigation

4.2.1 Encouraging uptake of HTTP/2 without transport layer encryption

- Facilitating development of Apache HTTPD module
- Facilitating development of nginx HTTP/2 module
- Investigation into development/extension of Internet proxies

Description:

Currently, HTTP/2.0 is specified by the 16th version of the IETF HTTPbis draft Ref [11]. This draft is approaching the state of final standard and the Working Group Last Call is set for November 2014. That is why many implementations are being developed and tested – see Ref [12].

Although HTTP/2.0 brings in plenty of improvements the protocol comes in two flavors: an encrypted version called "h2" and a clear text version called "h2c." Moreover, within "h2c", there are two mechanisms which provide clear text HTTP/2 communications:

1) "Upgrade" mechanism: starting from HTTP/1.1, one can negotiate switching to HTTP/2.0
2) "Direct" mechanism: directly start the HTTP/2.0 connection

Integration: Regarding implementations, both clients and servers are impacted. Several up-to-date implementations already exist, but they mostly implement HTTP/2 over TLS. However, a few implementations of the "Upgrade" and "Direct" mechanisms exist:

1) "Upgrade": ngHTTP2 (client, server), curl and libcurl (client)
2) "Direct": ngHTTP2 (client, server), node-HTTP2 (client, server), mruby-HTTP2 (client, server)

On the client side, these implementations only support a single HTTP request, which means one cannot automatically get the following requests for page elements like images, insertions, etc. So the use of these implementations on the user agents is quite restrictive. Therefore, server implementations are not deployed within commonly used webservers, limiting the widespread of clear text HTTP/2.

However, there are no well-known open-source webservers which support HTTP/2.0 (neither "h2" nor "h2c"). The two of the most common used webservers (Apache and NGINX) currently do not have projects in this area. Concerning browsers, Firefox currently, does not implement "h2c", nor does Chrome; and Microsoft is only considering to support clear text HTTP/2.0 in Internet Explorer or in Microsoft servers. Note, Microsoft and Google have already deployed HTTP/2.0 over TLS in most of their services.

Delivery: Actively encourage support of clear text HTTP/2.0, should help create an Apache module which supports HTTP/2.0.

The key is to ensure existence of the relevant open source code modules. There are a couple of options available to address the issue for an open source module being created as follows:
Alternative 1: One party with relevant expertise and resources provides such code.
Alternative 2: Operators jointly fund the 3rd party development of such open source code.

a) Raising project funds under the umbrella of the GSMA Web Working Group
b) Funding through the GSMA.

4.2.2 Object Level Encryption

Description: In order to protect the integrity and confidentiality of online transactions, HTTP traffic can be secured using transport layer security using HTTPS.

While HTTPS is widely used for ecommerce and banking and while there is a sense of understanding in the user community around the secure nature of HTTPS, using TLS and HTTPS for the majority of traffic creates performance and functional drawbacks mainly because the HTTP session is encrypted. Modern web architecture includes sophisticated caching schemes that involve fetching various objects (images, libraries, etc.) from various locations in the path to avoid latency and improve the overall user experience while reducing bandwidth use. This is an important aspect, especially in developing countries, remote locations and in general areas that lack fast network infrastructure.

The ability to atomically encrypt objects or even HTTP frames should allow content providers to avoid distributing their server key material across the network nodes and reduce the risk of compromising their security.

Network aware application optimization and management indirectly and positively affects the user experience and provides the access network with the ability to inform application owners and user agents about the health of the link to pre-emptively adjust the application behaviour without divulging any private user data. By anonymizing and randomizing this type of data, traffic can be less prone to fingerprinting.

Network aware info would be injected securely into the flow without compromising the integrity and confidentiality of the user transaction and can be decrypted by intermediary nodes, servers or user agents.

Atomic encryption of HTTP frames would also eliminate the need to circumvent the mixed content rules and same origin policies. These are current issues that content providers are facing when selecting between HTTP and HTTPS. These issues along with invalid certificate messages have made it very difficult for user agents to correctly and accurately inform users about security risks and lower the threshold for users that treat these serious warnings as exasperating messages.

There are several endeavours to address this issue with:

1) JSON Web Encryption Ref [13]
2) JWE allows to encrypt any kind of payload (as defined by JSON Web Signature)
3) Sub Resource Integrity Ref [14]

As the name suggests, the current specification only provides content integrity protection, not encryption.

Integration: These actions are not final solutions yet. The implications of integrating these options when they become available is unknown. However, only by looking at mechanisms like JOSE (Javascript Object Signing and Encyption) and Subresource Integrity, it may be concluded that likely additional implementation effort is required by content authors (e.g. in mechanisms similar to Subresource Integrity where content authors need to pre-calculate digests and add them to markup) or by parties operating web servers (e.g. in the case of JWE, where certain data objects require an encryption layer to deliver the encrypted objects).

4.2.3 Device and cloud based filters

4.2.3.1 Mobile operating system filters

Reference documentation:

1) Windows 8 Mobile: none for mobile (PC only)
2) iOS (iOS 7 onwards) Ref [15]
3) Android: none known

Description: Expose OS settings to restrict which applications can be used (including Web browsers), and which content can be consumed (e.g. explicit material, as understood by the OS vendor).

Integration: None for the network, as provided on the device OS and by the IS vendor cloud. However, an operator may wish to jointly promote such a feature with the OS vendor.

Notes: It is not clear how these settings manage encrypted Web content. For example, whilst it may be possible to block a Twitter app, it may not be possible to block access to Twitter URL (e.g. HTTPS://twitter.com), or to block certain paths within Twitter.

Mitigates the use cases: “Parental controls” but to be confirmed if this applies for encrypted content (as per note above).

4.2.3.2 Filtering apps (device-based)

Reference documentation: none

Description: An application installed by the customer or the device OEM. This will typically be OS-independent and require either installation on each network connected device or during manufacturing

Integration: None for the network, however many operators develop or champion high-quality filtering apps in their markets.

Examples: An Android application called Net Nanny Ref [16].

Notes: Since these apps reside on the device, there is less privacy impact if they are party to the content of encrypted communications. This is because the app client is already trusted by the user (i.e. it sees plaintext), and a filtering functionality within the app could be considered as a second user agent of that client, rather than an unauthorised intercepting party. There is a difference of opinion on this point and this can vary between operators and territories.

The effect is mostly limited to Android where the app has been granted root privileges to filter the HTTP and HTTPS stack.

For iOS: 3rd party apps that block or filter Safari traffic appear to be impossible with iOS apps due to the iOS sandbox restricting such privileges.

Mitigates the use cases: “Parental control”

Limitations: This solution likely works only for particular operating systems and therefore only for a subset of the devices available in the market.

4.2.3.3 Filtering browsers

Reference documentation: N/A

Description: The user is constrained to browsing via a specific ‘filtering’ browser.

Integration: Manual by user.

Example: AVG’s Family Safety solution Ref [17].

This utilises a configurable filter and monitor which may be set on the device or via the AVG PC client.

Mitigates the use cases: “parental controls”, but only for Web browsing via devices with OS-restrictions on which applications can be used.

Limitations: This solution only works if users are accepting to work with such dedicated filtering browsers. Most likely this is only a subset of the overall user group. The solution also requires that the OS can limit Web browsing to a particular browser; otherwise the filter can be bypassed by utilising a secondary browser app that may be installed by the user.

4.2.3.4 Filtering-apps (centralised)

Description: A cloud service to monitor and manage a group of family devices (Enterprise-facing solutions are not in the scope of this document).

Integration: Manual by user.

Example: Parents can register with a cloud based ‘parental controls’ service which they access from their PC. It can be used to configure access controls across their family’s mobile devices, including when to allow device access to the internet, blacklist enforcement, filters based on profane/sexual/violent content.

Notes: This solution seems to be only available for Enterprise use at the time of writing.

Mitigates the use cases: “Parental controls”

4.2.3.5 Filtering with per-service permission

Description: Within a given app or Web service, the user grants access to an external monitoring application to view the user’s activity and possibly block certain activities.

Integration: Manual by user

Notes: With Web services that implement a consent model such as OAUTH, it would be possible for the user to consent to their activity being accessible to an external service, allowing the external service to filter activity according to configured rules. However, sending all activity to an external service may introduce unacceptable latency, therefore an acceptable solution may require the social network to flag certain content and only inform the external service when attempts are made to access such content. This can be challenging to implement in practice.

Mitigates the use cases: “Parental controls”

4.2.4 TCP encryption without Authentication

Working documentation: IETF Working Groups are actively developing solutions for a new TCP protocol called TCPINC Ref [18]

Scope: Provide an alternative to TLS for encryption HTTP:// URIs traffic.

Status: The IETF created the TCP Increased Security (TCPINC) WG to specify solutions which protect TCP traffic from passive eavesdropping. The goal is to provide unprotected TCP connections with encryption solutions transparently such as opportunistic security.

The Working Group will develop extensions of the TCP protocol and new API to provide integrity protection and unauthenticated encryption of TCP streams.

Four individual propositions have been presented:

1. TCPcrypt Ref [19]:

   - Based on 2 new TCP options (CRYPT, MAC).
   - Key negotiation leverages TCP handshake + one ACK.
   - Encrypts and protects the integrity of the payload.
   - Protects the integrity of most TCP header fields.

2. Upgrade to TLS Ref [20]:

   - Proposes a new TCP option which indicates that the client desires upgrading the TCP connection to TLS using the legacy TLS handshake.
   - Does not encrypt the TCP header.

3. DTLS Ref [21]:

- A new TCP option (DTLS Record Protection) initiates the use of DTLS in modified Authenticated Encryption with Additional Data (AEAD) mode.
- Encrypts the content and protects the integrity of the TCP header.

4. TCP-AO Ref [22]:

- Extends the usage of the Authentication TCP Option (TCP-AO) to encryption.

Mozilla have made the 2nd (Upgrade to TLS) and 3rd (DTLS options) proposals to IETF HTTPBIS Working Group.

Mitigates against: All use cases blocked by the TLS server authentication.

Limitation: Does not apply to ‘HTTPS’ scheme traffic.

Advantages: Does not require user consent.

4.3 Managing ‘end to end’ HTTPS encryption

This section covers the use of ‘HTTPS’ URI scheme requests from a client to a content server, also referred to as ‘end to end’ HTTPS. It also includes ‘HTTP’ URIs that have been opportunistically encrypted, but only where this encryption involves the client and origin server. It does not cover HTTP URIs carried over a browser proxy secure channel (see the mitigation options covered in Chapter 3).

4.4 HTTPS today

4.4.1 What the network can know and mitigate

TLS Server Name Indication extension and 5-tuple collectively indicate:

Source	Layer	Provides	Mandatory/Optional
IP packet header	3 (network)	Source IP	M
IP packet header	3 (network)	Target IP	M
IP packet header	3 (network)	Protocol	M
TCP SYN message	4 (transport)	Source port, destination port	M
TLS ClientHello SNI (Server Name Indication)	5,6 (session, presentation)	Host (server.example.com)	O

This data can enable a coarse mitigation for the following use cases, accounting for the limitations of SNI and 5-tuple: **“**Regulatory filters”, “Parental controls”, **“**Customer QoE”, “MNO Customer insight”, ”MNO Network optimization”.

Use case	Limitation of SNI + 5-tuple
Regulatory filters, Parental controls	Per host (optional) or IP range only. Not suitable

	for ‘mixed content’ sites (family and adult content)
Customer QoE	Per host (optional) or IP range only. Requires network to keep up-to-date IP addresses of content sites to provide the best use experience.
MNO Customer insight	Per host (optional) or IP range only. Requires network to keep up-to-date IP addresses of content sites, with mapping to the associated domain.

In all cases the originating Source IP and Source Port may be shielded by a TCP bridge or other reverse proxy that conceals origin.

4.4.1.1 IPv6

In addition to the above,IPv6 traffic may also provide:

Source	Layer	Provides	Mandatory/Optional
Flow label to assist QoS (RFC 6437, RFC 6294)	3 (network) – IPv6 only	Delivery priority hint from server	Optional
Traffic Class to assist QoS	3 (network) – IPv6 only	Delivery priority hint from server	Optional

Limitations: These attributes are network-domain specific, i.e. the network will decide how or whether to act upon them. RFC 6437 highlights problems regarding the trust of sender flow labels (similar to the issue with DiffServ), as well as denial of service risk Ref [23]. If sent from a trusted server, and configured by the network, then these attributes can mitigate “Customer QoE”.

4.4.2 What the network may infer from heuristics

Heuristics supplements the SNI and 5-tuple data through matching the behaviour of an HTTPS session with previously tested behaviours.

Layer	Provides
3 (network)	Frequency of request/response
4 (transport) and 7 (Application)	Encrypted packet size
3 (network)	Active TCP connections to Source IP

The level of confidence for heuristics typically depends on whether the content provider has a single, or main, service, or whether it offers several services (and media types) simultaneously.

First case (single or main service): heuristic differentiation can be quite accurate, typically above 90/95%. This would be the case, for instance in these examples:

- Web browsing like any newspaper
- Email like outlook.com
- VoIP like Viber
- Video Streaming like Netflix
- Instant Messaging services like Whatsapp

Second case, provider with multiple services. Differentiating between web browsing and email for a provider that serves both is typically not possible. For VoIP, Video Streaming and Instant Messaging statistical patterns normally work with an accuracy >70%. But this depends on the relative importance of each service. For instance it is much more difficult to differentiate Facebook Messenger chats that go together with a huge browsing load, than Skype IM that goes together with Voice or Video.

4.4.3 Evolving encryption standards which may further hinder the network

4.4.4 TLS 1.3

The draft RFC for TLS v1.3 (Ref [24]) and mailing list discussions indicate that the TLS Server Name Indication extension may be passed encrypted. Whilst this decision has not been finalised, it would remove SNI visibility for servers that have upgraded to TLS v1.3, leaving network operators to determine the server host from the 5-tuple, which may be less accurate. Hence TLS 1.3 with encrypted SNI may hinder the network in coarse (domain-wide) mitigations.

4.4.5 Opportunistic encryption

Opportunistic encryption for HTTP URIs [Ref 25] describes how ‘HTTP’ scheme URIs, normally requested without encryption, can be accessed using TLS to mitigate passive snooping attacks. This is facilitated by use of HTTP Alternate Services [Ref 26] as explained in the document developed by W3C increasing HTTP Transport Confidentiality with TLS based Alternate Service [Ref 27].

In summary, a request to an unencrypted ‘HTTP’ URI can be routed via client-server negotiation to be served over TLS. A server software upgrade is required to support this. This creates encrypted transport but without server authentication. The intention is to raise the complexity of passive eavesdropping: sites requiring stronger security would continue to utilise a full TLS implementation with ‘HTTPS’ scheme URIs. Although associated with HTTP/2, it is not part of the core spec and is currently an experimental draft.

Opportunistic encryption does not require certificate authentication and therefore may be attractive for sites wishing to increase privacy without a significant cost. The network would still retain some visibility of the request based on the initial ‘HTTP’ request which initiated the alternate service flow.

4.4.6 QUIC Crypto

“QUIC is a datagram protocol, and the full payload of each datagram (above the UDP layer) is authenticated and encrypted once keys have been established” [Ref 28].

QUIC protocol does not use TLS for reasons explained in the QUIC Crypto document [Ref 29] – is because it is to improve on TLS efficiency, mitigate server IP address spoofing, and mitigate replay attacks.

One key point of QUIC is to reduce Round-Trip Time (RTT) to zero during connection establishment: i.e. the clientHello message will be immediately followed by the data request message without waiting for a server response. This clientHello message does allow a SNI (Server Name Indication), but first of all this is optional, and on the other hand, it does not require to pass the initial request (namely the first time when the client visits the server).

Therefore QUIC Crypto (at time of writing) will reduce Server Name Indication visibility for the network operator. It is feasible that an operator could match up the first client Hello message to a subsequent session to the same server, and hence infer the server name – but only if SNI was passed in the first client Hello message.

Other key points:

- Crypto is on by default; with the rationale not due to security but rather to ease end-to-end implementation. Google states in the QUIC document that traffic is going to be encrypted because it is the only method guarantees middle boxes passes the transmission when they try to filter or improve the traffic.
- It is unclear if and how QUIC Crypto can be disabled, because the handshake phase bundles the basic anti-DoS features that are key for the robustness of an UDP based protocol together with the 2-phase key agreement. This could be quite difficult to separate.
- End-to-end QUIC Crypto has a level of security equivalent to TLS with Perfect Forward Secrecy [Ref 30] cipher suite (e.g. very high); but with an aim of improved performance.

4.4.7 IETF TCPINC working group

Working documentation: IETF has a number of working documents on this research project [Ref 31]

Scope: Provide an alternative to TLS for encryption HTTP:// URIs traffic. In short, the payload is encrypted, but the server is not authenticated.

Status: The IETF created the TCP Increased Security (TCPINC) WG to specify solutions which preserve TCP traffic from passive eavesdropping. The goal is to provide unprotected TCP connections with encryption solutions transparently and massively deployable (opportunistic security).

The WG will develop extensions of the TCP protocol and new API to provide integrity protection and unauthenticated encryption of TCP streams.

The 4 individual propositions were presented during the IETF 91meeting in Toronto:

1. TCPcrypt
    - Based on 2 new TCP options (CRYPT, MAC);

- Keys negotiation leverages TCP handshake + one ACK;
- Encrypts and protects the integrity of the payload;
- Protects the integrity of most TCP header fields.

2. Upgrade to TLS

- Proposes a new TCP option which indicates that the client desires upgrading the TCP connection to TLS using the legacy TLS handshake;
- Does not encrypt the TCP header.

3. DTLS

- A new TCP option (DTLS Record Protection) initiates the use of DTLS in modified Authenticated Encryption with Additional Data (AEAD) mode;
- Encrypts the content and protects the integrity of the TCP header.

4. TCP-AO

- Extends the usage of the Authentication TCP Option (TCP-AO) to encryption;

The 2nd and 3rd proposals above proposals are developed by Mozilla.

4.5 Smarter encryption: proposals to help network visibility

4.5.1 Multiparty Content Integrity and Confidentility (MCIC)

The section on MCIC (also known as Object Level Encryption) describes methods whereby metadata pertinent to the network is made visible, but content payload is not.

Factors for success: The current integration method requires work at the content server to encrypt the different objects. This may be seen as more difficult than sending all content via a single encrypted tunnel. Therefore content publisher buy-in is likely to be a challenge, and will need to be encouraged by clear demonstration of the benefits (for example, the network will be able to provide improved customer experience).

IETF security advocates will need to be confident that any new standard does not weaken TLS. Rather that the network is allowed access only to metadata necessary for management functions. There may be pushback on certain network functions that are considered commercial (such as zero rating) as opposed to quality of experience related.

4.5.2 Attribute-Based Encryption

This approach differs from symmetric key encryption in that servers and clients do not need to share secret keys. A client can decrypt data only if its key is privileged to do so – as indicated by policies embedded in the ciphertext.

Attribute-Based encryption uses a Key authority: therefore it can be relevant for operator networks if the network is provided with a key allowing access only to certain information to allow routing, optimisation etc.

This is not yet standardised but there is a draft document developed for initial reference [Ref 32]

4.5.3 TCP Minion and SPUD

TCP Minion [Ref 33] and SPUD [Ref 34] propose improvements to TCP delivery, but thus far it is not clear if these protocols will impact network visibility of encrypted traffic. SPUD (‘protocol inside UDP’ has a goal to *‘expose minimum information to get midpoints to allow traffic’,* where ‘midpoint’ includes operator networks.)

4.5.4 Why would content servers implement these network friendly approaches?

Content servers would accept the cost of implementing solutions such as object or attribute based encryption so there needs to be a clear benefit to them. For both the network and the server, customer experience is a shared priority so there needs to be an understanding on how new approaches improve this.

An example could be the network being granted access to meta data sufficient to provide traffic management functions, whilst the server (and customer) retain control of privacy.

Certain deep packet inspection capabilities are lost to the network in order to maintain privacy and security. This may be a necessary compromise to gain support from content publishers to allow at least network traffic management and coarse filters to operate in an encrypted Web.

4.6 New transport methodologies and impact on network visibility

4.6.1 Multipath TCP (MPTCP)

Multipath TCP [Ref 35] (MPTCP) extends TCP to allow a TCP connection to use multiple paths. This enables a device to simultaneously use multiple IP interfaces to achieve better throughput and resiliency. For instance, a mobile device connected to a 3GPP network and one or more WiFi networks can use MPTCP to achieve better throughput and session continuity if one of the paths is disrupted. It is important to consider that the WiFi connections can be over carrier WiFi, public WiFi, or private WiFi networks, each controlled by different operators, none of which need to be the same as the one providing the mobile connectivity.

In order for a device to utilize MPTCP, there needs to be an entity in the network that can terminate the MPTCP flows. This could be the application server or an MPTCP proxy between the device and the application server.

4.6.1.1 MPTCP impact on network visibility, HTTPS traffic

In the case where the application server has inherent support for MPTCP, then the individual subflows can take different paths to reach the destination application server. This makes visibility into the complete TCP connection difficult since the subflows may pass through several operator networks. Therefore, any traffic identification and policy enforcement would need to be based on the five tuple information in the IP header. Service classification based on heuristics may not be possible.

In MPTCP the network is aware whether the TCP path is single or one out of multiple paths in the same TCP connection. This enables operators to take special actions on MPTCP.

In the case where an MPTCP Proxy is used, all subflows would terminate on the MPTCP Proxy on the device facing side and become a vanilla TCP connection towards the application server. If the MPTCP Proxy is controlled by the operator, then visibility of the TCP connections is possible.

4.6.1.2 MPQUIC additional implications compared to MPTCP, HTTPS traffic

The network cannot identify whether a QUIC stream is Multi-Path QUIC or just a single stream. This hinders separate actions on MPQUIC (like blocking).

MPQUIC also makes it easier to dynamically add or switch between connections, further complicating e.g. heuristics.

4.6.2 Multiplexed HTTP/2 over TLS

HTTP/2 , as with SPDY, promotes multiplexing of multiple streams over the same TCP connection. This is likely to have an impact on heuristics systems utilising TLS Server Name Indication extension and 5-tuple, because these reveal information about a single TCP connection.

For example, if “socialnetwork.example.com” serves a range of Web pages, VoIP, video streams, Instant Messaging sessions and photo galleries, they may wish to multiplex a user session to reduce the number of required TCP connections. If they utilise a reverse proxy to aggregate these content types, then even a server name indication or 5-tuple will only reveal the domain of the reverse proxy. Heuristics will need to distinguish the streams and match each to a known behaviour pattern.

The impact on network visibility depends on the network heuristics capability whether such content aggregators will multiplex streams for mobile connections, given that packet loss in one stream will impact the TCP connection as a whole, and radio networks exhibit higher packet loss than fixed connections.

4.6.3 QUIC

As described above, QUIC (Quick UDP Internet Connection) utilises QUIC Crypto and not TLS. The impact on network visibility is reduced Server Name Indication input, and hence more reliance on lower-layer 5-tuple in heuristics.

4.6.4 STS

The Sub-Transport Session (STS) layer is a recent proposal that has been made around the recent IAB SEMI workshop (IAB Workshop on Stack Evolution in a Middlebox Internet, HTTPS://www.iab.org/activities/workshops/semi/) and advocates for a session layer on top of UDP that would allow endpoints cooperate with middleboxes in the path in two ways:

- By exposing session establishment and lifetime semantics to each device along the path.
- By letting endpoints determine which middleboxes along the way perform which modifications to provide which services.

Each packet-modifying device in the path would be STS-aware and would return a report on how it modifies packets and why, allowing endpoints to make decisions on the characteristics of their desired connection and establishing direct collaboration with the network elements.

4.6.5 Bandwidth negotiation between client and network

4.6.5.1 AEON at IETF

AEON (Application Enabled Open Networking) was discussed informally at the last IETF meeting. This topic has not yet become an official work item. There is more work justification required to prove that AEON aims to improve on mechanisms such as DiffServ to allow feedback between application and server.

4.6.5.2 CDN interface API and architecture development

Today, major CDNs including Limelight and Akamai can agree to host and serve the content publisher's server certificate on their behalf. This is to allow both the SSL handshake and content serving to be performed by the CDN, and requires a business relationship and security assurances from the CDN to protect the content publisher's key.

This relationship can be extended to include the network operator. Since the content publisher, CDN and network all want to deliver the best customer experience, the CDN content cache can be hosted within the network. This can provide the CDN and content publisher with fast hosting close to the network edge, also offering the cache with network context to improve QoE. In return, the operator should be able to regain visibility of HTTPS CDN content requests in order to apply necessary traffic management . This would likely be a bilateral (or trilateral) business solution and therefore no standards can be recommended, however there may be an associated case, namely CDN interconnect [Ref 36]. This involves CDN A serving CDN B's content, because CDN B does not cover a particular territory. If the content is served over HTTPS, then CDN A will need to be a trusted party to the SSL certificate which CDN B hosts on behalf of the content publisher. This is therefore a business and technical option, to be investigated individually by operators and their CDN partners.

4.7 Other factors affecting HTTPS uptake

4.7.1 Uptake of HTTPS by content servers

The rate of uptake of HTTPS by content servers will be dictated by:

1) Increased customer demands for privacy
2) Cost of implementation
3) Performance impact
4) Publisher content being shared and served via HTTPS social networks

Options 1) and 4) are established today. However, options 2) and 3) are likely to accelerate due to the decreasing cost of certification, with basic certificates available for free from certain providers. Improved performance is also being pushed by major Web players via HTTPS://www.istlsfastyet.com service. 'HTTPS as a Service' providers (see below) aim to offer both reduced cost and improved performance to stimulate HTTPS uptake.

4.7.2 HTTPS ‘as a service’

Cloud providers such as CloudFlare and Amazon CloudFront are offering to manage the ‘heavy lifting’ of SSL certification. They host and provide the certificate on behalf of the content website, and Google PageSpeed is likely to follow (it is currently in a limited trial). The added benefit to the website is that these providers can also optimize their Web pages as part of the service.

Further testing is required to confirm, but it seems like the certificate can indicate the content publisher SNI rather than that of the Cloud host (although that may not be supported by older browsers)

4.7.3 Mixed content

Certain major sites (including online newspapers) do not utilize HTTPS encryption on their webservers due to the issues this causes with 3rd party JavaScript (including adservers and URL shorteners). However, they will often use HTTPS for article comment login, to protect the privacy of users from politically sensitive countries.

4.8 Summary and recommendations for end-to-end HTTPS

4.8.1 Collective action on end-to-end HTTPS

Operators are encouraged to collectively support these options via the GSMA:

1) Continue focusing on long term technical mitigation options (as listed above).
2) Collaboration with industry organisations where it is sensible and meaningful.
3) Investigations into smarter encryption proposed for an informal meeting at IETF 92 in March 2015, either through direct attendance or contribution to preliminary objectives.
4) Engagement with ‘aggregating’ Web sites such as Facebook to have early visibility of any plans for HTTP/2 multiplexing over TLS, and formulating technical recommendations accordingly.

4.8.2 Individual operator action

Individual operators are encouraged to investigate

1) The heuristics offerings of their network vendors as regards their support for HTTPS end-to-end traffic, including multiplexed HTTP/2 over TLS
2) Discussion with CDN partners regarding hosting of CDN HTTPS caches in-network.

Annex A Categorisation of Use Cases

Key to table

1. Requirements: Based on the use-cases at Ref [1]
2. Potential mitigation(s): for that use cases, including solutions under investigation
3. Existing mitigation for encrypted traffic: shows what can be achieved today, and any limitations
4. Source use-case: References the original use-cases from the “Proxy User Stories”. “New” indicates a new use-case/requirement.

Use Cases	Requirements	Source use cases
MNO Customer insight	MNO customer insight from data traffic	New
Customer malware protection	Customer malware protection	"Francine's Virus Scanner"
Customer QoE	Improved customer QoE and data cost savings.	"Xavier’s Data Saving App" and New
MNO network optimization	1) Reduced network infrastructure cost 2) Cell optimization of data traffic	"Darlene's Mobile Network", "Wallace's network traffic flow"
Regulatory filtering	Apply regulator blacklist per-territory (where applicable)	New
MNO Smart pricing	Smart pricing, including zero-taxation, of data traffic	"Peter’s Flowers delivery company" and New
Parental controls	Kid safe-surf and access to content depending on remaining data plan quota	"Nancy's kids' mobile devices"
3rd party service augmentation	1) Providing subscriber information to 3rd parties (authentication and "data plan status") 2) Protecting subscriber privacy	"Quincy’s Online Movie download store", "Liam's Mobile Identity Proxy" and "Mike's Music Service"
MNO Services	DNS routing within the network to an operator customer portal (e.g. mobile top up)	New

4.9 Use-cases out-of-scope for this paper

Category	Use-case	Reasons for outside MNO scope
Censorship circumvention	Ulrich's Censorship Circumvention	Fixed network (Other tools or manual proxy config)
Filtering/censoring	Alex's Enterprise-Supplied Laptop	Enterprise fixed network use-case
Filtering/censoring	Bobbie-Sue's BYOD at Work	Enterprise fixed network use-case
Filtering/censoring	Henrietta's Jailhouse Schoolroom	Enterprise fixed network use-case

Filtering/censoring	Ian's Compliance Mission	Enterprise fixed network use-case
Filtering/censoring	Kirk's Kids	Home fixed network use-case
Filtering/censoring	Oscar’s Certification Company	Enterprise fixed network use-case
Filtering/censoring	Vicky's School and Library Internet Access	Enterprise fixed network use-case
Filtering/censoring & Malware protection	My home is my castle	Home fixed network use-case
Home network proxy discovery?!	Jane's Thing	Home fixed network use-case
Improved QOE	Students' Shared Cache	Shared fixed network infrastructure use-case
Improved QOE	Tom's Rural Broadband	Shared satellite network infrastructure use-case
Improved QOE	Eliot's Village Internet Connection	Shared fixed network infrastructure use-case
Network authentication	Charlie's Coffee Shop	Enterprise fixed network use-case
Network based app debugging	Grant's Debugging Proxy	Fixed network use-case

Annex B Document Management

B.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	11/09/2014	*New Position Paper for publication*	ENCRY group	Kevin Smith (Vodafone), Istvan Lajtos (GSMA)
1.1	3/10/2014	*Updated content of Position Paper for publication*	ENCRY group	Kevin Smith (Vodafone), Istvan Lajtos (GSMA)
1.2	5/01/2015	*Updated content*	ENCRY group	Istvan Lajtos (GSMA),/ Kevin Smith (Vodafone)
1.3	28/1/2015	*Updated and fresh content reflecting current status*	ECNRY group	Istvan Lajtos (GSMA, Kevin Smith (Vodafone)
1.4	12/2/2015	*Updated content based on feedback from ENCRY team*	ENCRY group	Istvan Lajtos (GSMA)

B.2 Other Information

Type	Description
Document Owner	Web Working Group – ENCRY sub-group
Editor / Company	Kevin Smith Vodafone

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com