IN THE INVESTIGATORY POWERS TRIBUNAL **IPT/21/37C**

Mr [redacted] and others

v

National Crime Agency

Expert report of Professor Ross Anderson

I, Ross John Anderson, will say as follows.

1. I am Professor of Security Engineering at Cambridge University and also at Edinburgh University. I am a Fellow of the Royal Society, the Royal Academy of Engineering and the Institute of Physics. I won the Lovelace Medal, the top UK award in computing, in 2015.

2. I have worked or consulted for a wide range of technology companies both before becoming an academic and since, including IBM, Microsoft, Intel, Google and Samsung. I have long and wide experience of communications systems, location systems, payment systems, monitoring systems, and security, in both industry and academia. My CV can be downloaded from my web site[1].

3. My relevant background experience ranges from both technical and policy work on the interaction between cryptography and law-enforcement access to communications, through research into cryptography and security, to engineering work on designing secure systems. I wrote the standard textbook, *'Security Engineering – a Guide to Building Dependable Distributed Systems'*, which can also be found on my web site.

4. On the technical side of cryptography, I have been active in the design and analysis of cryptographic algorithms and protocols; a pioneer of the study of hardware tamper-resistance; and a pioneer of the study of the economics of information security. I have also consulted extensively for major technology companies. As examples, I worked on the STS specification for prepayment utility meters, now used in over 400m meters in over 100 countries; on the design of HomePlug AV, which is used in WiFi relays

[1] See http://www.ross-anderson.com or http://www.cl.cam.ac.uk/~rja14

worldwide; and on the design of Android Pay which enables hundreds of millions of people to make credit card payments with their mobile phones.

5. My research work in recent years includes examining how Android security mechanisms can fail[2] and how the Signal encryption software can be embedded in a newspaper app to protect sources who contact journalists. As a result, I have some familiarity with the cryptographic, access-control and other security mechanisms employed both by chat apps and by Android phones.

6. I am the Principal Investigator of the Cambridge Cybercrime Centre, which collects large quantities of spam, phish, malware and posts to underground crime forums. We license our data collections to over 100 researchers at over 50 universities worldwide; most of our licensees do academic research on cybercrime. We also provide threat intelligence on malware to organisations including the FBI and NCSC (GCHQ).

7. On the side of technology policy, government attempts to limit the civilian use of cryptography have led to conflict both with the IT industry and with civil society. Since the early 1990s I have been engaged as both an academic and as an industry consultant in analyses of various government initiatives, from proposed encryption standards in the 1990s through the debate over access to encrypted iPhones in 2015 to recent attempts following the Investigatory Powers Act to compel communications service providers such as Facebook to add backdoors to messaging systems.

8. I set up the Foundation for Information Policy Research (FIPR) with colleagues in 1998. FIPR was active in educating MPs during the passage of the Regulation of Investigatory Powers Act and subsequent surveillance regulation, and the later Investigatory Powers Act. We also worked with other European NGOs to establish European Digital Rights (EDRi), the main digital rights NGO in the EU. I am also on the Advisory Board of the Electronic Privacy Information Center in the USA.

9. When the Investigatory Powers Commission was set up, I was one of half a dozen representatives of civil society invited by Sir Adrian Fulford (as he then was), the first Commissioner, to run training sessions for him and for the Judicial Commissioners.

Instructions

10. I have been instructed by Middleweeks, who represent the [redacted], in conjunction with Kenyon McAteer and Avisons.

11. I have been asked to consider the material provided to me and provide an opinion on, inter alia:

    a. The method of exfiltration of the material, including both metadata and message contents, obtained by the JIT from the EncroChat system.

[2] See Laurent Simon and Ross Anderson, 'Security Analysis of Android Factory Resets' and 'Security Analysis of Consumer-Grade Anti-Theft Solutions Provided by Android Mobile Anti-Virus Apps', both at Mobile Security Technologies 2015, and also available from my web page

b. The decryption of message contents.

c. The creation of the json files and provided to the NCA by the JIT via EuroPol.

d. The interception of photographs.

12. I have been provided with seven suspected EncroChat handsets and asked to conduct an experimental exploration of them.

13. I have also been asked to consider the reports drafted by Luke Shrimpton, who appeared as an expert witness for the prosecution in R v [redacted] et al., and comment on his conclusions, where relevant.

Previous involvement & material relied on in making the report

14. The material provided to me for the preparation of this report is listed in Annexe B. Most of the references in this opinion are to page numbers in the Preparatory Hearing Bundles relied on before Dove J in [redacted], [redacted] and the subsequent appeals. Where references are to material from other sources I have made this clear.

15. I was instructed by Middleweeks Solicitors in the Crown Court case R v [redacted] and others where I was supplied with the Operation [redacted] documents and technical material disclosed by the Crown. I listened to parts of the [redacted] and [redacted] hearings before Dove J over the court video link.

16. I was called on to analyse the [redacted] data at short notice before the initial hearing in R v [redacted]. The files are large with multiple copies of data from different phones, scripts of different vintages, and images of docker containers that can be used to run the scripts. I therefore recruited as an assistant one of my research students, Mr Nicholas Boucher. He worked entirely under my direction to help me wade through the masses of undocumented data that we received and to replicate one of the network experiments of the NCA's Luke Shrimpton. Nicholas Boucher has worked for me again in this instruction for the IPT, which has given us even larger quantities of data plus phones for forensic analysis. I have also been assisted by Dr Sergei Skorobogatov, a Senior Research Assistant at Cambridge University whose speciality is hardware security. He has helped me attempt to extract memory contents from EncroChat devices. Both have worked under my direction at all material times. I have also had helpful discussions with Avisons Solicitors' technical expert, Mr Kushinder Raheloo. However the conclusions herein are mine.

17. This report is based on analysis of the material disclosed in Schedule B, the NCA evidence in the pre-trial hearings into A B D & C and [redacted], and the results of the limited experiments we have been able to perform with the handsets since we finally got one working on December 17th. I undertake to inform my instructing solicitors and the court if we discover further material evidence.

18. The case R v A B D & C relied on an interpretation of the technical evidence that was set out in the Campbell report, which was mistaken in several important ways.

a. It assumed that the use of the Signal protocol with its end-to-end encryption meant that communications could not be intercepted en route from one handset to another.

b. It deduced that the collection was entirely from endpoints; in effect that an implant installed by the French and Dutch Joint Investigation Team in thousands of EncroChat phones simply sent to the Gendarmerie copies of messages sent, messages received, locations, photos, passwords and notes.

c. It noted that, during the live intercept phase of Operation Emma, messages were sometimes intercepted within a tenth of a second of the user pressing the 'send' button, argued that waiting for this message to be written to the Realm database on the handset and reading it from there would take too long, and deduced that the live collection took place from the random-access memory (RAM) of the handset.

d. It assumed that the collection or interception was performed by an implant installed in the device as a rootkit rather by modifying the EncroChat app.

e. It further assumed that the encryption or decryption of messages was a single atomic operation performed by the sending handset, and reversed by the receiving handset.

f. It further assumed that all the collected material was already present in the memory of the handsets, and it merely fell to the implant to passively copy data from the device memory and forward it for collection.

g. It further assumed that the material collected from the handsets was simply passed on to the Gendarmerie's C3N server, from there to Europol, and from there to the NCA, subject only to filtering by country and triage for immediate threats to life.

h. It followed that the legal arguments should focus whether collection from the RAM of endpoint devices should count as copying or interception.

19. Assumptions (a), (b), (c), (d), (e), (f) and (g) are not consistent with the evidence.

20. The evidence shows that the operation undertaken by the French and Dutch Joint Investigation Team (JIT) was a much more substantial and complex enterprise which involved modifying the EncroChat app, duplicating the server infrastructure maintained by EncroChat at the OVH data centre in Roubaix, and installing a load balancer so that messages to and from the customer devices could be copied to the JIT servers or diverted to them so that the EncroChat servers did not see them. Access to message contents involved decryption. Finally, the copies of messages and their metadata supplied in the evidence bundles were probably extracted from a database maintained by the DGSI that aggregated data from multiple sources, both handsets and servers, rather than coming entirely from the handsets directly.

21. The only reasonable description for this activity is interception – as indeed was claimed by Europol in its media release of July 2020.

22. As the Court of Appeal relied on the Campbell report's interpretation entirely in A B D & C and to a significant extent in Atkinson [redacted] et al., I shall explain in detail why the picture presented in that report – and relied on subsequently by Dove J and by the Court of Appeal – is both inaccurate and misleading.

23. I will first describe how crypto phone systems evolved and how they work. My purpose is to explain the goals of crypto phone vendors and the design constrains under which they work. This will lead to an explanation of how the encryption works in chat apps at three different levels – local, transport and end-to-end – which were confused in the earlier hearings. I will then analyse the goals of the Joint Investigation Team and the design constraints under which they in turn had to operate thanks to the security architecture of an encrypted chat system.

The evolution and operation of secure messaging systems

24. Modern secure communications systems typically rely on a mechanism called public-key cryptography whereby each principal generates a private key, with which they can decrypt messages sent to them, and derives from it a public key, which others can use to encrypt those messages. It is not feasible to derive a private key from the corresponding public key. The effect is that if one user, Alice, wants to speak to another user, Bob, she can look up Bob's public key in a directory, encrypt her message, and send it to him. Bob can then use his private key to decrypt her message.

25. During the 1990s, the US and UK governments used various policy mechanisms including export controls to discourage the use of strong cryptography between end users, fearing that this would damage intelligence and law-enforcement capabilities. Industry insisted that strong cryptography was needed to protect electronic commerce. One of the resulting accommodations was that commercial email services such as Gmail, Hotmail and Yahoo do not by default encrypt messages all the way from the sender to the recipient. The Transport Layer Security (TLS) protocol that they use encrypts the traffic from the user's web browser only as far as the webmail service. The messages are stored in clear on the mail server, where a law-enforcement or intelligence agency can access them with a warrant.

26. However, a third party monitoring the user's Internet connection cannot intercept their mail, as the transport-layer encryption protects all the data in the user's session with the mail server, including not just the message body but also mail headers such as the "To", "From" and "Time" fields. The TLS encryption thus protects a UK user of an open Wi-Fi network in a hotel from having their Gmail read by someone else in the hotel. It can also protect a UK Gmail user in a hotel in Shanghai from having their Gmail read by the local Ministry of State Security; for this reason, Gmail is often restricted or blocked in countries like China and Iran.

27. The disclosure by Edward Snowden in 2013 of the scale of surveillance by the NSA, GCHQ and other Five Eyes agencies alarmed potential targets including journalists, lawyers, human-rights workers and business leaders. It led not just to the Investigatory Powers Act and the establishment of IPCO, but to a major effort by equipment vendors and service providers to improve privacy technology.

28. Messaging services that support end-to-end encryption (E2EE) became widespread during 2013–15; the leaders now include WhatsApp, iMessage and Signal. The effect of such encryption is that service providers can no longer provide police and intelligence agencies with unrestricted access to the contents of messages or calls in response to interception warrants.

29. Modern messaging services such as Signal typically use encryption for three different purposes:

    a. The message contents, such as "See you at 7", are encrypted end-to-end to ensure they are available to the sending and receiving devices only, unless a user elects to back up their messages to an online service.

    b. The encrypted contents are accompanied by the metadata that instruct the message server what to do with it. At the very least this means "From:" and "To:" but it typically includes more. For example, many messaging services support a "burn time" after which the message is deleted. These metadata are protected by transport layer encryption, usually TLS, between the device and the servers which contain the users' mailboxes. Depending on the system design, there may be separate servers for pictures, and for users to back up their historical messages online (hence the need for burn times to be in the metadata rather than just bundled with the message contents).

    c. The messaging app uses a local file encryption mechanism to protect messages, pictures and other data stored on the phone. Modern phones enable storage keys to be linked to specific applications, so that only the Signal app can decipher stored Signal messages, only the WhatsApp app can decipher stored WhatsApp messages, and so on.

30. Encryption may be used for other relevant functions. For example, the long-term storage in a phone is typically encrypted using a key that can only be accessed by entering the correct unlock password. This is used on the phones supplied by EncroChat, along with a further password to unlock the EncroChat app itself. Operation Emma therefore collected passwords that were typed into infected EncroChat devices, enabling the police to unlock captured phones.

31. The options open to an agency that wants access to cleartext messages include:

    a. Seizing any backups made by users to an online service;

    b. Seizing equipment from one or more users;

    c. If a seized device is locked, compelling the user to disclose the password;

    d. Using the law to compel the suppliers of messaging software or services to provide back doors that give some means to decrypt messages;

    e. Using some combination of legal compulsion and technical interference to compel the use of weak cryptography;

    f. Modifying the software on one or more of the devices where the messages are stored, whether user devices or servers, and making it copy messages;

g. Modifying the software on one or more devices and causing it to choose weak keys, leak the keys, or use keys controlled by the intercepting agency;

h. Attacking other parts of the infrastructure, such as the servers on which users' keys are kept.

32. Common consumer electronics equipment and telecommunications services have features designed to thwart one or more of these attacks.

a. Attacks via equipment seizure are made more difficult in phones and laptops by encrypting their storage using a password, a biometric or a combination of the two. This has led to an arms race between forensics firms who invent tools to unlock phones, and phone vendors who block them.

b. Laws that enable government access to keys have been the subject of policy tussles for over 25 years. The Investigatory Powers Act s253 provides ministers with the means to force vendors with a UK presence to give access; but users can buy overseas products instead.

c. The collection of communications using malware is mitigated by multiple mechanisms that make it harder to make unauthorised modifications of device software, that detect new strains of malware by their behaviour, and that update phone software in response.

d. Key stealing by software modification is also blocked other measures. Modern applications change keys frequently so that compromised keys can decrypt the current message, but not either past or future traffic.

33. There are complicated trade-offs between protection mechanisms on the one hand and product features on the other. For example, some messaging systems (such as WhatsApp) keep traffic logs to analyse which users communicate with which others, so that advertising can be targeted, with the side-effect that the police can find out who messaged whom and when. Some systems (such as WhatsApp and iMessage) encourage users keep backups of their old messages online, so that they can recover from the loss of a phone, but then the police can recover messages too. In short, messaging is a complex ecosystem, and more detail may be found in standard works such as my 'Security Engineering' textbook[3].

The evolution and operation of crypto phone systems

34. Since 2014 various 'crypto phone' products have come on the market that avoid these trade-offs and promise uncompromising protection. They typically offer enhanced resistance to forensic analysis, in that a captured phone cannot be read out using standard forensic tools. They also offer enhanced resistance to traffic analysis; a common feature is that users are not identified by their phone numbers but by a user ID or 'handle' such as '[redacted]'. Some vendors make it difficult for

[3] '*Security Engineering – a Guide to Building Dependable Distributed Systems',* Wiley 2001, 2008 and 2020

governments to serve warrants, whether by sheltering in an uncooperative jurisdiction or simply not publicising an address for service.

35. Not all crypto phones are aimed at criminal markets. The pioneer was Blackphone, whose vendor Silent Circle sold phones to special forces, corporates, high-net-worth individuals and human-rights workers.

36. Even where a phone's users are mostly criminals, not all of them are; I am told that the working EncroChat phone supplied to us for analysis came from a phone reseller.

37. There is also no clear divide between crypto phones and off-the-shelf products. Apple's iPhone resists forensic analysis if locked; Signal and Wickr offer handles other than phone numbers; and Signal frustrates warrants by keeping no logs.

38. Users of communications systems need to be on the same system to talk to each other. As a result, the markets for communications systems and services tend to have dominant firms. For example, over 40 social-networking systems were launched in the early 2000s but one of them, Facebook, became dominant.

39. In the case of crypto phones, the pattern has been for one system to attract a lot of criminal customers, and then be taken down by law-enforcement action. Users then migrate to another system and the pattern continues.

    a. The initial market leader was Ennetcom, until it was raided and closed down by the Dutch and Canadian authorities in 2016[4]. The Dutch police reportedly stole PGP keys from a server in Canada as part of the takedown[5].

    b. PGP Safe was killed by the Dutch National High-Tech Crime Unit in 2017[6].

    c. In 2018, Iron Chat was shut down, again down by the Dutch[7].

    d. Also in 2018, the US, Canadian and Australian agencies closed down Phantom Secure, whose CEO had boasted of selling to the Sinaloa cartel[8].

    e. The 2020 takedown of Encrochat in an operation by the French and Dutch authorities led to the present case.

---

[4] C Osborne, "Dutch police close Ennetcom encrypted communications network", ZDnet, Apr 25 2016

[5] See "Dutch Police Fight Crime by Cracking PGP Phones", Arnout Jaspers, Communications of the ACM, July 10, 2018

[6] K Alam, "Dutch Police Seize Another Company that Sells PGP-Encrypted Blackberry Phones'', Cybrary, May 11 2017

[7] D Goodin, "Police decrypt 258,000 messages after breaking pricey IronChat crypto app", Ars Technica, Jul 11 2018

[8] K Bolan, "Richmond IT expert sentenced to 9 years in U.S. prison for helping violent criminal organizations", Vancouver Sun, May 29 2019

f. In March 2021 the French, Belgian and Dutch authorities took down Sky ECC. The Europol press release said, *'The operation is an essential part of the continuous effort of judiciary and law enforcement in the EU and third countries to disrupt the illegal use of encrypted communications, as was already displayed last year following the successful de-encryption of the EncroChat communication platform."*[9]

g. In June 2021 the US and Australian authorities took down the An0m messaging app. That was a Trojan-horse operation from the beginning: the An0m phones designer became an FBI informant, and the phones were marketed to criminals by an informant of the Australian Federal Police.[10]

40. The EncroChat phones we have examined are modified Aquaris BQ X, 2X or 3X ('Carbon') devices. They use the Android operating system, which is 'open-source' in the sense that anyone can download the code, modify it as they please, and install it on handsets. As well as the 'normal' Android, there is a hidden partition, accessed using a second password, containing the actual EncroChat app. The handsets also have their firmware modified to block the USB port and thus frustrate normal forensic tools. In the photograph below, I exhibit on the right a normal Aquaris X2 purchased on eBay, and on the left a phone believed to be supplied by EncroChat. The normal phone has a 'bq' seal on top of the screw holding down the USB port cover, below the battery on the picture; but on the suspected EncroChat device, this has been tampered with. The usual firmware mechanisms for putting a phone into emergency download (EDL) mode for forensic examination have been blocked, so to experiment with the phone we moved the motherboard of the Encro device to the unmodified device, and added a physical switch to force a hardware EDL boot.



[9] Europol, 'New major interventions to block encrypted communications of criminal networks', Mar 10 2021

[10] Tim Swanston, "How the AFP and FBI cooked up the idea of AN0M over a 'couple of beers'", MSN News, June 8 2021

Intercepting a phone system's servers

41. The attack on EncroChat was launched via the EncroChat central service. The French Operation Emma was formally started on November 5th 2018, after the takedown of the Phantom Secure system mentioned above. The authorities saw captured EncroChat handsets talking to a server at the OVH data centre at Roubaix near Lille in France, and a warrant was issued on Dec 21 2018 to copy the server. The French lead investigator Jérémy Decou was asked '*if I was involved in obtaining images from the servers*' and '*if I had gone to OVH to get copies of the data*'; he replies '*Yes, with the magistrates' authorisation and I obtained copies from the server.*'[11]

42. Later warrants[12] describe an architecture whereby dozens of linked services ran in virtual machines (VMs) or containers on a server in the OVH Roubaix datacentre.

43. Virtualisation allows a single physical server to run VMs or containers on top of hypervisor software which separates them from each other, just as the Android operating system software separates the apps on a mobile phone from each other. The VMs or containers operate as separate computers. This gives real benefits in terms of power efficiency, maintainability, and resilience. For a more complete description of access control and virtualisation, see my Security Engineering textbook.

44. VMs are often protected against casual snooping by data centre staff using encryption, with the noted mechanisms `LUKS` and `TANG` being well-known[13]. But the data centre operators can take images of VMs for law enforcement or intelligence agencies, who can study these images to find the cryptographic keys they need to decrypt the VMs and to break authentication[14], so that they can masquerade as one VM to another and thus perform both passive and active interception against VMs in a cluster.

45. For convenience I will refer to the VMs or containers running on the server or servers at OVH and supporting the original EncroChat service as the 'Encro cluster'.

46. The Encro cluster's encryption and access-control mechanisms were defeated by an engineer from the IRCG (Institut de Récherche Criminelle de la Gendarmerie), the French police forensics team. They discovered not just services to support text, voice and multimedia communications, including a message server, two media servers and a server to support voice calls, but business process support including key management, journal analysis, user support, SIM management, a private network server for configuring BQ devices, SIM allocation to devices, and management of resellers. A subsequent forensic analysis by the NCA found 71 VMs[15].

---

[11] Witness statement of Jeremy Decou MG11 of Sep 25 2020; pp 1176–85

[12] pp 1569–1575

[13] p 1286, p 1299

[14] p 1571

[15] Derrick Avery, 'Operation Venetic – Initial Analysis and Examination Report', item 280, 04/03/2021

47. In addition to the systems found at any other communications service provider, there was an update server. Normal Android phones receive software updates, for both the operating system and apps, through the Google Play Store. Its associated service manages push notifications, such as telling a messaging app that a message is available for collection. Such services could provide several ways for law-enforcement and intelligence agencies to attack a messaging service, so it was rational for the EncroChat operators to provide their own.

48. The NCA has images of these systems but has not provided them to the defence[16].

49. To sum up, Encrochat was a phone company – a public telecommunications service – not just selling handsets but running the same kind of infrastructure that one might see in a mainstream communication service provider. The goal of Operation Emma was to take it over surreptitiously, monitor all the traffic for two months, and then close it down. Furthermore, Operation Emma collected data that a legitimate communications service provider would not, including logs of the locations of all handsets; these were actively collected by identifying the closest cell towers, and there was also an attempt to collect the names (SSIDs) of nearby WiFi hubs that appears to have failed[17].

50. I note that while the NCA worried about the legality of an implant collecting data on nearby WiFi access points, and had the warrant re-issued to cover this, they did not worry that the implant was collecting location data in the form of cellsite locations. This had been explained by Jérémy Decou[18] but its significance may not have been properly considered.

Intercepting metadata and message contents

51. I am asked to explain exactly how the French and Dutch authorities may have intercepted or collected the message traffic sent to and from Encro handsets.

52. It took many months of work to understand the Encro cluster and devise an attack, which exploited or duplicated VM components of the server system at Roubaix. The French eventually had access to everything: not just the update server for the handset software, and a backup server on which customers' notepad data were stored, but also the server that managed cryptographic keys[19]. The NCA officers record 'seizure of development server', which appears to imply access to the source code and development environment for the Encro apps on the handsets and perhaps also for the

---

[16] Derrick Avery, 'Operation Venetic – Initial Analysis and Examination Report', item 280, 04/03/2021

[17] In the [redacted] evidence bundles, we found 557 instances of the 'dumpsys wifiscanner' command being logged. None of them have any reasonably expected output; they are all either blank or contain the text "error: wifiscanner offline".

[18] p 1743

[19] French warrant, p 1571 ff; translation 1576–1582

code used to customise Android (e.g. to set up the hidden partition and to block software access to EDL mode)[20].

53. The Dutch have long practice at taking over crypto phone systems, as noted above, and we learn that the exploit was developed jointly with the French. The IPCO decision records that the implant was created by the JIT, rather than by France[21].

54. By January the Gendarmerie had established access to the EncroChat messaging server. As explained above, its communications with subscribers' phones were encrypted using TLS, and it is standard practice in messaging systems (as in email systems) to use 'certificate pinning', which means that the phones would always encrypt their communications with the server using the TLS public key in a specific certificate[22]. The JIT had access not just to this public key, which may be embedded in the phone software, but also to the corresponding private key, which would have been kept at the EncroChat server. Both keys were supplied to me later so that I could repeat Mr Shrimpton's experiments.

55. An email at the time by Luke Shrimpton[23] notes *"... it looks like the French are planning some significant activity utilising their access to the EncroChat servers. We suspect it will... do something that allows them to use their intercept on the server to decrypt any data that passes through it*."

56. It is worth thinking through what it means for a law-enforcement agency to have an intercept on a messaging server but still be unable to break the end-to-end encryption of message contents. Luke Shrimpton's report 445 documents what interception looks like when the EncroChat app is run with a test message server, and I reproduce below part of one of the tables, which illustrates an intercepted test message.

    Lawyers may be familiar with the idea of writing to a client in jail by sending the letter inside a larger envelope addressed to the postmaster of the prison, who will check that the prisoner is allowed to receive unopened mail from them, and if so will deliver it. Extending this analogy, one might imagine Alice putting her message in a small envelope, then writing on the cover "From Alice, to Bob" and putting it in a larger envelope that she posts to a trusted postmaster Sam. Sam opens the outer envelope, checks that Bob is on Alice's "friends' list", and if so places the inner envelope into a fresh larger envelope that he sends to Bob. Thus only Alice and Bob see the message; Bob gets mail only from trusted correspondents; and only Sam knows that Alice is even in communication with Bob. Outsiders can only see that Alice and Bob are both in communication with Sam.

    To continue our analogy, Eve the eavesdropper must find some way to get past both envelopes. She might either read the messages in Alice's office or Bob's, before or

---

[20] Wayne Johns case book 19 Jan 2020 – Europol meeting notes

[21] p 8

[22] Or to a specific combination of subject name and certificate issuer

[23] Luke Shrimpton to Greg Elliot, 24th January

after transmission, or she might bribe or hack or coerce Sam, and then find some way to make the inner envelopes transparent.

Action	Intercepted Data
Alice sends message to Bob.	'<message id="**7ad65af6-0ada-4b9f-a0b1-faf8754574c4**" to="bob@encrochat.com/ec664" from="alice@encrochat.com/ec664" type="chat"><body>IwohBY/IVWeACEDwL+dC4Dpin+OhAOHRmw+3MBXe9PKup0YaEAAYACIFOSVOdiMCB5lKSORnFg</body><request xmlns=\'urn:xmpp:receipts\'/><burn xmlns=\'urn:xmpp:burn\' delay=\'604800\' /><whisper xmlns=\'urn:xmpp:whisper\' whisper=\'2\' /></message>

57. Formally, a cryptographer might use the standard notation $\{A\}_K$ to denote the message A encrypted by the key K, and describe a message from Alice to Bob as

    A –>S: {Message N, From Alice, To Bob, $\{$message body$\}_{KABN}$, Burn time T$\}_{KTLS}$

    where the TLS key KLTS encrypts all the metadata plus the message contents, which have already been encrypted under the end-to-end message key KABN – the Nth key used between Alice and Bob. To make this double encryption more visible, we might colour the data encrypted in the outer, transport-layer, envelope in blue and the data encrypted in the inner, end-to-end envelope in red:

    A –>S: {Message N, From Alice, To Bob, {message body}$_{KABN}$, Burn time T$\}_{KTLS}$

58. In technical language, an intercepting agency must first obtain the TLS private key or otherwise obtain the data inside $\{\ \}_{KTLS}$, then it must also either obtain the end-to-end message key KABN or otherwise obtain the message body inside $\{\ \}_{KABN}$.

59. The format of the intercepted message is extended markup language or XML, in which angle brackets of the format <tag>data</tag> are used to tag data items. The end-to-end encrypted message contents, which I described symbolically as {message body}$_{KABN}$, appears in the actual EncroChat message as <body>Iwoh...RnFg</body>.

60. The TLS private key compromise gave the JIT access to traffic data, so they could track who was communicating with whom. But its value was limited by the anonymity of the phone users and the encryption of message contents. Two more tasks therefore remained:

    a. Adding location data so that phones can be mapped to individuals via cellsite co-location analysis or other methods such as from nearby Wi-Fi signals;

    b. Obtaining the message keys so that the message contents could be read.

61. The JIT attacked the handsets from April 1st by sending to the EncroChat handsets a software update that enabled their location to be tracked and their message contents to be read. This was described by Jérémy Decou to the NCA as a '*rogue app*' that would collect chats, device locations and cryptographic keys[24]. The description was later changed by Emma Sweeting to '*An implant within an Application will be placed on all EncroChat devices worldwide ... via an update from the update server in France*'[25].

62. If the source code of the app was modified, as appears to be the case, then her use of the term 'implant' was inaccurate. Her language suggests a third-party virus hooked into an executable program; but if the program itself was modified to do the work, that is more simply understood as a material change to the telecommunications system itself.

63. We read that the JIT identified and copied both the development server and the update server. The former presumably had the EncroChat source code and tool chain, with which it would have been straightforward to develop and test a modified version of the EncroChat app. As for shipping the modified software to the handsets and getting them to install it, the EncroChat operators were using their own update server rather than Google's, but presumably used the same mechanism, as it is near universal: software updates are authenticated by means of a digital signature. The signer has a private key that they apply to the new version of software to produce a signature, while the existing software has the corresponding public key and will accept an update only if it carries a signature that can be verified using that public key. As the French analysts were able to extract the private key for the TLS encryption protecting the traffic between the handsets and the message server, they should have been able to extract the signing key and forge a valid signature on their rogue app update.

64. The next problem facing the JIT would have been how to deal with the sheer volumes of material generated by the bulk exploitation of thousands of phones. We hear that they observed over 60,000 registered devices and initially guessed there might be

[24] p 1743

[25] p 1231

50,000 in use; this was cut to 30,000. The NCA estimated 9,000 in the UK with 4,500 "*operational at any given time*"[26], but later discovered 5402 handsets in use, generating intercepts of 9.6m messages and 200k images in the first 14 days of the operation[27]. Worldwide, there was an enormous amount of message traffic to be intercepted – tens to hundreds of thousands of customer messages a day, plus about ten times that many location messages. Furthermore, in the initial week of the operation, the plan was to collect and upload from each updated device its stored messages, photos, contacts, and cryptographic key material, in the 'phase 1 sweep'. The JIT were aware that "*their network deployment will lead to significant data draw across network, potentially knocking users offline" (ibidem)*.

65. Bulk exploitation of the EncroChat devices and traffic was also constrained by the need for it to be covert. It would have been rash for the JIT to send the collected / intercepted traffic directly to a Gendarmerie server, as assumed in the Campbell report. Although a normal smartphone communicates with all sorts of services associated with its vendor and its apps, many of the EncroChat devices had been configured to not run normal apps or make normal phone calls. Extraneous communication would have been abnormal and could have sounded an alarm.

66. The ingenious solution was to install a load balancer in OVH Roubaix which intercepted the traffic from the EncroChat devices to one or more of the EncroChat servers. The technical work of installing the load balancer and its supporting systems was carried out by the Direction Générale de la Sécurité Intérieure (DGSI), France's equivalent of our Security Service[28]. This resulted in the technical details remaining classified under French law. DGSI is even more averse to publicity than our Security Service; its headquarters is blurred out on Google Street View[29]. It has historically been even more opposed to the civilian use of cryptography, which has been much more restricted in France than elsewhere in Europe. Indeed Operation Emma originally aimed to arrest the EncroChat system operators and the Lille warrants cite the French offence of '*provision of a means of cryptology that does not exclusively ensure authentication or integrity control functions without prior declaration*'[30].

67. The JIT / DGSI also largely duplicated the EncroChat cluster as I will discuss later.

68. A load balancer is a class of device commonly used in data centres to accept incoming traffic and distribute it between multiple servers for processing. For example, if web traffic to a retailer's website is ten times the maximum load a single server can cope with, the retailer might have ten servers and have a load balancer send queries to them in turn. Real systems are often more sophisticated. It is common for an application

---

[26] p 1383; warrant p 1399; also Luke Shrimpton, note of NCA meeting, 25 February 2020

[27] A Cochran, warrant review, p 1892

[28] pp 1282 (French) and 1288 (English)

[29] 84 Rue de Villiers, Levallois-Perret, France

[30] E.g., page 1284

load balancer to decrypt incoming TLS traffic, work out what service is required, and encrypt it again if needed while redirecting it to the appropriate service.

69. Given that the JIT / DGSI already had both the EncroChat public and private TLS keys, such a device offered a neat solution to several problems at once:

    a. They planned to send location update messages so that the phone users could be identified and tracked. These messages were voluminous; about ten location messages for each customer-to-customer message. A load balancer could intercept them before they get to the genuine EncroChat cluster so they did not raise an alarm there.

    b. Similarly, the bulk data collection during the first-phase sweep after device exploitation also created massive quantities of data and a real risk of detection. Intercepting this before it hit the genuine EncroChat cluster would have been critical for maintaining covertness.

    c. The load balancer could provide a covert command-and-control (CnC) channel between the Gendarmerie and the handsets. Such a channel is often a weak point in exploits. When malware is discovered on a machine, a standard countermeasure is to identify the addresses to which it 'phones home', and then circulate these on worldwide blacklists so that any machine that also communicates to those addresses can be identified and examined. By running CnC through the EncroChat cluster itself, and using the load balancer as a proxy, the JIT neutralised some common anti-malware defences.

    d. The load balancer could operate either in passive mode – merely taking a copy of every message that passed through it – or in active mode, to interfere with passing traffic.

    e. As it was co-located with the message server, a load balancer could operate quickly. Campbell's analysis shows that in many cases the JIT / DGSI intercepted messages within a tenth of a second of the customer pressing the 'send' button.

70. The fact that the collected or intercepted material was exfiltrated via the servers at Roubaix was well known to the NCA. In the warrant application, DG Lynne Owens tells us that the implant would be unable to transmit any further data to the JIT once the servers were taken down [31].

71. This fact is further confirmed by a court judgment in Norway according to which investigators gave evidence that they had monitored communications during the live phase of Operation Emma, revealing a single encrypted data stream destined for a single IP address at Roubaix[32].

---

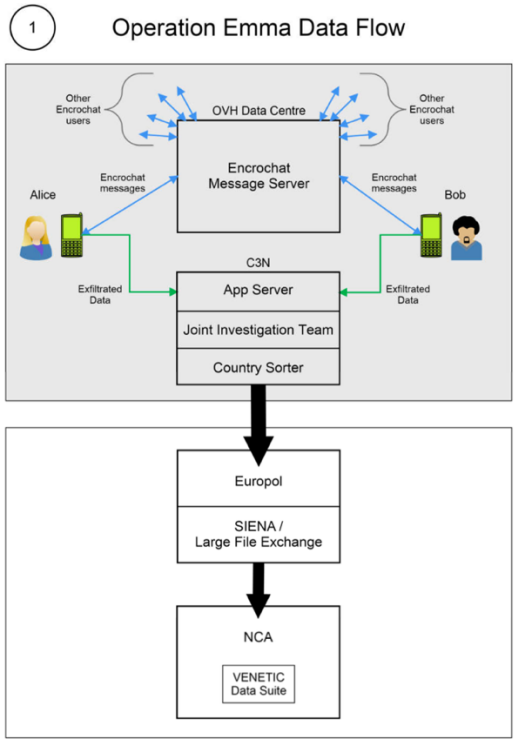
[31] p 1398

[32] Oslo District Court, 21-030329MED-TOSL/05, 02/11/2021. Based in part on this evidence, the Norwegian prosecutors submitted that the EncroChat material provided by the JIT was taken from the servers and not the handsets.

72. In preparation for the attack, the JIT / DGSI replicated the EncroChat infrastructure. As noted, this was not just a message server; there were also servers for images, for update, for device keys, perhaps to support voice calls, and much else. We read "*The set-up is quite complex because around 96 virtual machines need to be installed in order to replicate the infrastructure in the EncroChat*"[33].

73. The Gendarmerie got a warrant on March 4th to lock the Domain Name Server (DNS) for encrochat.ch and its subdomains. DNS translates human-readable names such as encrochat.ch to the Internet Protocol (IP) addresses of the machines providing the service. The effect of this lock was to ensure that the EncroChat operators could not escape the interception by moving their service to a data centre elsewhere[34].

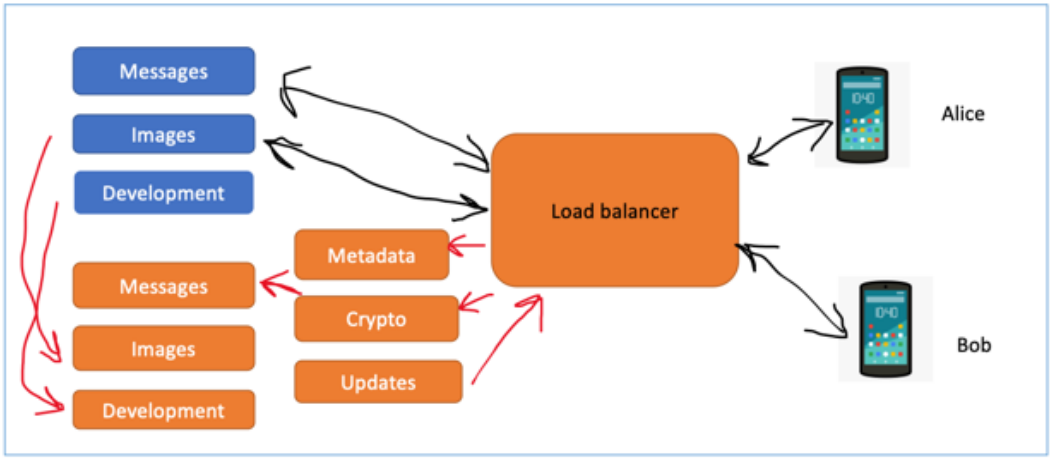
74. The Campbell report was therefore quite wrong to state that the data were exfiltrated from the infected phones directly to the Gendarmerie's C3N server. The illustrative diagram in the Campbell report, which I reproduce below, shows two users, Alice and Bob, exchanging messages via the EncroChat server while their handsets reported to a separately located C3N server. This misconceptualisation led to incorrect analysis by Campbell and later by the courts. It was an unforced error as the IPCO warrant is quite clear on the matter. The French warrants are also quite clear that the DGSI controlled the interception.



[33] PH 3329, paragraph 1(a)

[34] We discovered that an EncroChat phone we set up on a private network would only get its DNS data from the fixed IP address 8.8.8.8, a service operated by Google, which would have taken as its authoritative source the domain administrator against which the court order was issued. Once we understood this, we devised a way to override it for a network that we controlled.

75. The design of the EncroChat system required all the messaging and control traffic to and from the EncroChat devices to pass through the EncroChat cluster in Roubaix, and this naturally led to an attack where the traffic to and from the exploit was intercepted there too. The phones were programmed to contact the domain encrochat.ch, which now pointed to the load balancer that could redirect traffic to any EncroChat service to a copy service controlled by the JIT. The phones were also programmed to use transport-layer encryption tied to the EncroChat TLS public key, for which the JIT also had the private key – so they could intercept the traffic at the load balancer, to which the encrochat.ch domain and its subdomains were locked by court order. The 'collection mechanism' referred to in the Lille warrants (e.g. 2 April, p 1155) consists of a modified EncroChat app, the load balancer, a substantial duplicate of the EncroChat cluster, and other services. The correct diagram is thus more as follows.



76. Here, the original EncroChat system is coloured blue, and the components interposed or duplicated by the DGSI are coloured orange. The handsets' software has also been substantially modified by the JIT / DGSI.

77. This explains how the Phase 1 downloads and later the location messages could be intercepted before reaching the EncroChat cluster without raising a alarm. It may also explain why, at the Europol meeting in February, Luke Shrimpton made notes, "*Is the DB dumps intercept?*" "*Data sent to OVH server*", and "Looks like it's intercept"[35]. If the Campbell architecture had been in use by the JIT, that question would never even arise. But given the actual architecture, even the "DB dumps" are encrypted at the handset under the EncroChat TLS public key and sent to its infrastructure at Roubaix, where they are intercepted by the load balancer.

78. The NCA view seems to have swung back and forth. Clare Meehan had already reported to the CPS on 23rd January that 'the UK are likely to treat this material as TI'. The view might have changed following an email that Jérémy Decou sent to

[35] Notes of Europol meeting, 19–21/02/2020, p 3314

Emma Sweeting on 29 January saying, "*You want to know what our rogue app can collect, here are the main features: extraction chat (text and media) + AES key database; extraction of notes database; extraction cell ID to find which country the phone is*". Sweeting had replied "*our main question (if you are able to assist) is whether the messages are collected from the databases stored in the devices or in live time from the server. This will greatly assist us in deciding how we could use this in the UK*."[36] These descriptions do not explain the interception architecture or clarify the distinction between phase 1 and phase 2, so it is understandable if NCA officers were confused about these issues prior to the Europol meeting in February.

79. After that took place, we find in the casebook of NCA officer James Wilmott, pages 07–09, dated 20–21/02/2020, that "From the information provided forward looking data will only be domestically collected in France via the server as opposed to targeting every individual device."

80. The testimony of Emma Sweeting about how she concluded at the Europol meeting in February that a TEI warrant would suffice, as her Gold Commander had requested, appears in the transcripts of the preliminary hearing of R v [redacted].

81. As for the detailed architecture of the phase 2 live interception, when Alice sends Bob a message, it does not just flow from Alice to the messaging server (Sam) to Bob. It now flows from Alice to the load balancer, which we might call Louis, to Sam, and then once more to Louis, and then to Bob.

82. This suggests one possible explanation for the curious fact that incoming messages appear to be collected twice. For example, in the received message quoted on pp13–14 of Mr Shrimpton's June 3rd report, the message body appears seven lines from the top on p13 – and again five lines from the bottom on p14. A quick search discloses many other examples in the disclosed data. They look as if an incoming message from Alice to Bob, as it appears in Bob's evidence bundle, has been intercepted by Louis en route from Alice to Sam, and a second time en route from Sam to Bob. I will give a concrete example of this message duplication later and then discuss the options for interpreting it.

83. Our next question is the architecture of the exploit. Normal apps on Android are restricted by Android's access control mechanisms to read and write their own data; access to system resources requires the user to grant the relevant permissions. App access to cellsite location data involves the getAllCellInfo system call which would require the ACCESS_FINE_LOCATION permission. However, if 30,000 EncroChat devices had simultaneously asked their users for this permission, it would be obvious that something was wrong.

84. The simplest solution is for the modified app to acquire 'system privilege', also known as 'running as root'. This means that the app finds some way to escape the Android access controls and run as if it were part of the operating system. It can then use any resource it wishes, without having to ask the user for consent.

[36] pp 1736 ff

85. That the modified app somehow did this is evident from the fact that it also collected device unlock passwords, as system privilege is required to capture these passwords. It may also explain the records of attempted WiFi scanning, which mention 'dumpsys wifiscanner'. Dumpsys is an Android command-line utility to read the local WiFi status, typically invoked by a developer via the Android Debugger (ADB), so the full command would normally be 'adb dumpsys wifiscanner'. If the modified app has acquired system privilege, though, the 'adb' may be superfluous.

86. The IPCO warrant refers to an 'implant', which as we noted is language introduced by Emma Sweeeting. In theory there might be three possibilities.

    a. The first is that the app used a vulnerability in Android to assume system privilege. This is by far the simplest; the modified app could have called a service in the JIT cluster to download a known Android vulnerability, run it and then attempt to use system privilege to see if it had worked. Alternatively, the vulnerability could have been packaged inside the modified app. That said, downloading an exploit when needed is better from the viewpoint of covertness, anti-forensics, and compatibility across various devices.

    b. The second is that the updated app downloaded a separate implant which used a vulnerability to assume system privilege and then remained in the devices' operating system as a rootkit.

    c. The third is that the update server remotely exploited the devices, installing an implant that ran as root, while leaving the Android app unchanged.

87. The Campbell report and the subsequent court hearings assume that option (c) was the chosen route, or just possibly (b).

88. Option (c) is excluded, because message burn times have been disabled on the EncroChat X3 in our possession. This establishes that the attack did indeed modify the app.

89. I consider option (b) to be significantly less likely. As the EncroChat devices had their hardware and system software extensively modified to make forensics harder, an off-the-shelf 'government Trojan' implant might have been tricky to adapt.

90. It is both difficult and time-consuming to reverse-engineer app code and understand it. Even though a separate implant could in theory have interfered with key management routines to weaken the cryptography, or even read cleartext messages out of the device's memory, that would be a huge amount of extra work. The original EncroChat app bytecode, supplied to us to replicate Luke Shrimpton's experiment, has also been obfuscated to make reverse engineering even harder. An attacker with access to the development environment, as the JIT / DGSI did, would not have to do this extra work as they could just modify the app. Option (b) would therefore have added needless complexity to an already complex and time-critical operation.

91. Option (a) would thus have been much easier, and is supported by Jérémy Decou's statement that his 'rogue app' would get cellsite data (although, as he admits, he is an investigator not an engineer, so his statements cannot be entirely relied on as technical descriptions). Getting an app to run as root by exploiting a vulnerability in the

operating system is not difficult. Many Android vulnerabilities are discovered, patched and published, and most handsets are vulnerable to published vulnerabilities most of the time[37]. The EncroChat handsets are likely to have been particularly vulnerable as they did not use the Google Play Store to update Android regularly. We are told that that the JIT managed to exploit the BX2 handsets from the outset and then, later in the operation, the more modern BX3 (Carbon) devices too. This is consistent with the app downloading and trying one vulnerability, then another.

92. We have not yet been able to establish whether the modified app was the EncroChat app itself, the 'push' app which handled communications with the chat server[38], or both. For present purposes it appears unlikely that the distinction is important.

How the message contents could have been decrypted

93. The French, Germans and Dutch described the operation as interception, along with Europol. The NCA however, after the Europol meeting in February, described it as 'forward facing collection'.

94. Before the live interception started, there was a 'phase 1 sweep' after the modified app was installed on each phone. It collected saved messages (which were by default stored for seven days) as well as saved pictures, contacts and cryptographic key material, and sent them to the DGSI.

95. We are told that the messages harvested from the handsets when the modified app was activated are in the `encrochat.realm` file which decrypts to `encrochat.realm.decrypted` using the key provided, which appears to have been harvested at the same time. In the case of the dump that has IMEI 358098093388140 and is associated with the username [redacted] (which is alleged by the NCA to be a phone used by the claimant [redacted], this is found in the `Raw` folder for April 2nd. Updated copies were harvested on April 6th and 8th; they decrypt with the same key. Copying the database and its decryption key to the load balancer would have been straightforward for the EncroChat app, which creates and manages this material anyway. Even if it did not yet have system privilege, it would still have had sufficient access.

96. Realm is a public-domain software tool for database management and was a natural enough choice for the EncroChat app developers to store messages. In our work for the R v [redacted] case, we found we could examine the messages and other data therein using a standard tool, Mongo DB Realm Studio. Since the IPT data was too large, we used the Node.js Realm SDK rather than the Realm Studio for the data analysis in the IPT case. In practice, this means we relied on a combination of both tools in preparing

[37] D Thomas, AR Beresford & A Rice, 'Security metrics for the Android ecosystem', SPSM 2015 – Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (2015).

[38] p 1243; see also Shrimpton, 'EncroChat: Message Status and Packet ID', 13/08/21, [redacted] case

this report. The use of Realm by the EncroChat app is confirmed by Luke Shrimpton's experiments reported on June 3rd.

97. The Realm files do not contain lock screen passwords or geographical information, whether cellsite or Wi-Fi, which appear to have been collected only during the live 'phase 2' of the operation. This is consistent with root access being acquired at the time of the attack for surreptitious cellsite data collection, as well as for lockscreen password capture.

98. The French warrant authorised the exploit on April 1st; there are a couple of messages from that day remarking

    a. "2020-04-01T15:22:29+00:00","2020-04-01T15:22:45+00:00","[redacted]","[redacted]",604800,"**Is your phone playing up**","",""
    b. "d01f94bf-0ca2-49a3-a40c-007626f31774","2020-04-01T15:22:46+00:00","2020-04-01T15:22:57+00:00","[redacted]","[redacted]",604800,"**Mine going a bit slow**","",""

99. The data offered to the courts as evidence, in the various csv files of messages and folders of images, is produced by Luke Shrimpton's python scripts from:

    a. the `encrochat.realm.decrypted` database in the case of historical messages (messages collected or intercepted at the time the implant is ordered to do an initial ingest); and

    b. the file `json_data` in the case of the messages intercepted afterwards, along with geolocation data and passwords. (This also contains pointers to files exfiltrated from target devices such as Realm databases, and files probably exfiltrated from EncroChat servers such as photos.)

100. The data that was intercepted live appears in the files `json_data` which we find in the `Raw` folders for all the daily captures. The records include both messages and location data together with passwords and pointers to further data such as pictures. In addition, intercepted images appear in the `Raw` folders.

How message contents could be decrypted

101. The next question is how the message contents could be determined. Given remote and surreptitious access to the handsets, there are many ways in which this could be done.

    a. One possibility is a variant of the Campbell hypothesis: that each handset just takes a copy of each message sent or received and sends it to the load balancer. This would involve sending twice as many messages, and if the copied messages were as verbose as they appear in the json files, perhaps 3–4 times as much data.

    b. A second possibility would be for the handset to merely send the message contents; everything else is already available to the JIT via the load balancer.

This would still mean twice as many messages but less than double the data, so it would be more covert.

c. A third is for each handset to send to the load balancer every key that it generates. Again, this would mean double the number of messages, but only a small increase in data volume.

d. A fourth would be to send the key along with the message. For example, the message ID sent with each message might contain the key. This was the approach of the Escrowed Encryption Standard, also known as the Clipper chip, promoted by the US government during the first round of tussles over cryptography policy in the 1990s. A more recent example is given by the An0m network, taken down on June 7–8 2021, where according to unsealed court documents the "*FBI… built a master key into the existing encryption system which surreptitiously attaches to each message and enables law enforcement to decrypt and store the message as it is transmitted.*" [39] The EncroChat app could encrypt each message key under a key known to law enforcement and use the result as the message ID.

e. Another time-honoured exploit technique is to undermine the random-number generator that is used to generate keys (and the message ID). Cryptographic libraries typically contain a random number generator designed so that an adversary cannot determine any of its outputs even given access to many earlier and later outputs. But the random number generators provided with computer programming languages are much simpler; each random number they generate is a well-known function of the last few. Given a few samples, an analyst can determine all the past and future outputs[40].

102. My assessment in June was that (e) above was the most likely. Simon Csoka QC therefore put this method of decrypting messages to Luke Shrimpton in cross-examination and he agreed that it was possible. Mr Shrimpton then argued that such an arrangement would have been fragile, in that decryption would fail if synchronisation were lost. Yet this is exactly what we see in the data. The interception does indeed appear to have been fragile, with only perhaps 20–40% of the total traffic for the period appearing in each of the evidence bundles.

103. I will first describe the underlying technical reasoning, and then an experiment we conducted to investigate whether extra messages were sent by an infected EncroChat phone to a message server.

---

[39] Operation Trojan Shield Court Record, US District Court, Southern District of California, case 3:21-mj-01948-MSB Document 1, at https://www.documentcloud.org/documents/20799201-operation-trojan-shield-court-record, p8

[40] There was a scandal when the US National Security Agency (NSA) got the US National Institute of Standards and Technology to standardise the 'dual elliptic curve random number generator' which appeared to be secure but wasn't, enabling the NSA to break systems that used it, and later also allowing foreign intelligence agencies to break US systems too.

104. In the Signal protocol, a ratchet mechanism is used to generate a new key for each message or small group of messages, with the properties of forward and backward security. The intent is that the compromise of a device will not compromise its future or past traffic. The handset maintains a 'ratchet' of key material for each counterparty, and at each exchange of messages, a key exchange is performed whereby each device generates an ephemeral private-public keypair and updates the ratchet using its own private keys and the other party's public keys. The ratchet is so called because the process is one-way and cannot be wound backwards. The mechanism has evolved over time and has added complexity to deal with users who have multiple devices, messages that arrive out of order, and so on[41]. The effect is that to decrypt to the traffic between two parties, an adversary must either maintain access to one of their devices, or have access to one device at some time and thereafter know all the keys generated by that device.

105. In the case of EncroChat, the ratchet for each communicating pair is maintained in the Realm database, which also contains the user's long-term private key. These ratchets were all collected in the phase 1 sweeps and are provided in the Further Material bundles. Given this access, the cryptanalyst would only need to know the output of the random number generator used by each communicating party in order to determine the keys. So the modified EncroChat app would use a predictable random number generator rather than the secure one provided with the Signal software.

106. As noted above, the technical work of interception was carried out by the DGSI, France's domestic intelligence agency, the equivalent of MI5[42]. I would expect any intelligence agency to create a decryption capability given that it was open to them to do so. It is a long-established standard operating procedure, given the many benefits; these range from operational covertness through to access to other material, such as the ability to decrypt voice calls[43]. The French have been adamantly insistent for decades on their right to break any cryptosystem used for confidentiality. In the early days of mobile telephony, mobile phone calls could not be encrypted at all in France; and as noted above, Operation Emma was started not to catch drug dealers but to close down an illegal cryptosystem. As also noted above, the Operation Emma activity was described as decryption (and as interception) in most of the non-UK publicity around the case.

107. When it came to making a best guess between options (a) through (e) a significant factor appeared to be the intermittency of the decryption. In each evidence bundle, we find intercepts for a minority of the days of the operation. What might be causing these gaps?

---

[41] There was a significant evolution in the period 2013–5 when the protocol was adopted by WhatsApp and extended to support its functionality.

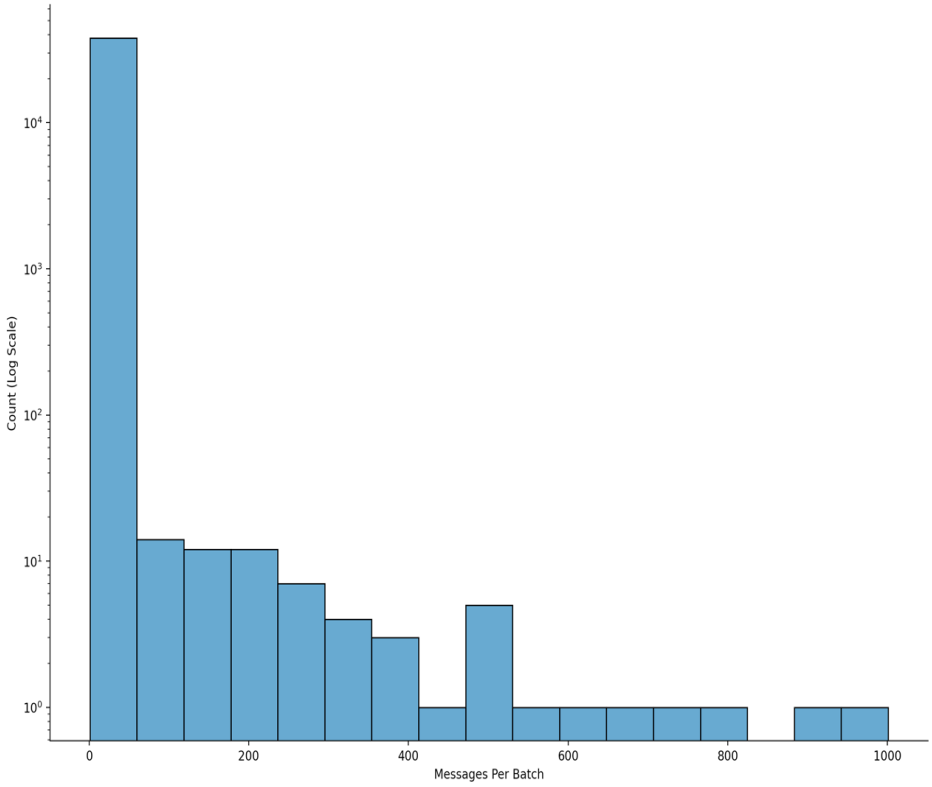
[42] pp 1282 (French) and 1288 (English)

[43] Voice intercept was done by the French in this case; Riaz Jakhura, email of 23 January

108. Let us now consider the evidence around intermittency in more detail.

a. Looking at the Raw data from the phone alleged to be [redacted]'s, we find at timestamp 1590072057.419377 on May 21st a whole batch of messages. This is the batch referred to by Mr Shrimpton at the bottom of p12 of his report of June 3rd. When we format it according to the standard JSON specification, that one line turns into 28,302 lines, which contain 494 messages from the 20th and 21st May and no other types of data.

b. The following graph plots the number of individual messages included on a single line for all of the json_data files provided as evidence in Operation [redacted] and Operation [redacted]. Lines containing non-message data only (such as location) are omitted. The vast majority of messages were received by the JIT as individual messages, but some arrived in quite large batches.



c. A natural question is whether these batches arise as a result of queuing in the handset, or at the load balancer, or elsewhere. We considered the DGSI server, but this seems unlikely as the gaps are not correlated between handles. We considered traffic queues at the load balancer, but that would have brought down the entire EncroChat system.

d. We learned from Luke Shrimpton that the EncroChat messaging server is an MQTT broker, and we verified this by repeating one of his experiments. MQTT may have been chosen by the EncroChat developers because of its support for 'topics'. One device (say a handset) can publish a message against a topic, and another can subscribe to it.

e. When something publishes against a topic, the broker builds a queue with all messages in that topic until a subscriber pulls them. When we ran the disclosed .apk code in simulators with an MQTT server, we observed that each receiver is treated as a topic; this approach gave the EncroChat developers a pre-built central messaging server. Did a failure to clear the queue on the police copy of the MQTT server perhaps give rise to the observed message batching? We therefore looked at the 'no-data' days in the evidence in the Operation [redacted] and Operation Subzero bundles, but could see no pattern.

f. We appear to be left with the hypothesis that the data drop-outs during the live intercept phase are caused by loss of synchronisation which prevents decryption.

109. Some support is given to this hypothesis by the fact that during April 1–8, loss of synchronisation appears to be fixed in some phones by a new phase 1 sweep. Some users had as many as five phase 1 sweeps. These collect, among other things, the ratchet values from the Realm database for every one of the phone's corresponding parties. It is a reasonable inference that after April 8th, the JIT / DGSI implemented a lightweight phase 1 sweep that collected ratchet values and other crypto key material as needed from time to time.

Experimental exploration

110. We set up an Aquaris BQ X3 (Carbon) phone supplied to us by Avisons Solicitors on an isolated network to communicate with an emulation of the EncroChat message server and activation server, following the setup in Luke Shrimpton's experiment which we had repeated in the R v [redacted] case. We were advised that this phone had belonged to a reseller rather than a criminal defendant, and that the reseller claimed the phone had never been used in crime. Test messages were sent and the resulting ciphertexts observed. The messages did not result in any additional message traffic on port 8886, suggesting that if any of options (a) through (c) were used then the exfiltration was not done via the message server, or even using the MQTT protocol on which the original unmodified EncroChat app relied.

111. This experiment does not rule out other means of exfiltration including a dedicated command-and-control server using a different protocol. It also does not rule out the possibility that the modified program was designed to exfiltrate data only for a fixed period of time, or that the exfiltration functionality was killed at the end of the interception period by means of a command from DGSI. We have not observed any location messages, but as a lab precaution we removed the phone's SIM card to prevent a kill command from reaching it.

112. As it is not clear that all X3 phones were modified by the exploit, we verified that our sample phone was indeed running the modified app by sending messages with a one-day burn time, and observing that they were not in fact burned 24 hours later. We also observed that an existing message on the phone had been there for many months despite a 7-day burn time. This confirmed that it had a modified app.

113. These network experiments rule out a number of options but do not yet give a clear explanation of how location data, possibly keys, and possibly messages, were exfiltrated by the modified app via the load balancer.

114. We are planning further experiments to explore how this might have been done. However if the NCA maintain that all the message contents and metadata were exfiltrated directly, the court might reasonably ask the NCA expert to explain and demonstrate the method, given their much greater access including to images of the servers at Roubaix, multiple captured devices, and their existing network simulation expertise demonstrated in Luke Shrimpton's experiments reported in June 2021.

115. The NCA understood the data to be supplied by the JIT well enough for Luke Shrimpton and colleagues to write the scripts to analyse it in advance of its arrival once the interception commenced. They were even supplied with test data[44]. Yet they failed to disclose this documentation to the defence or explain the data disclosures in other than telegraphic ways. Furthermore, the scripts initially supplied to us had comments removed, which had the effect of making them harder to understand. As luck would have it, we discovered a version of the code in one of the docker containers that still did have the comments, as well as what is described as 'slightly speculative' code for undeleting files[45]. There is even a comment 'Will it have any part it our arguing about timelines?' showing that at least some of the processing was done with a view to presenting a particular interpretation of events to the courts.

Where the json files were created

116. Even before the network experiment, we had significant doubts about the Campbell hypothesis that the evidence files, called `json_data` in the bundles, were generated by the implant by passive observation of the phone's memory and passed on untouched to the NCA. The provenance of these files goes not just to whether the material is collected or intercepted, but also to its provenance and thus evidential reliability. If these files were indeed created on the device with a view to being used in evidence, why not just sign them in the device itself to preserve them as untampered evidence?

117. We have no real idea what software wrote that the `json_data` file, let alone a copy of it. Yet it is this program that generates the bulk of the evidence on which the prosecution relies in the EncroChat cases. But in terms of where it ran, was it

a. the modified app, or

b. a script that extracted data from a database compiled by DGSI or the JIT from intercepted material and potentially other material too?

118. Judges in the criminal cases have so far found for (a) following the Campbell report, taking the view that each line of the JSON file is a faithful reproduction of a

---

[44] Crown's On-going Response to Defence Disclosure Requests – Document Number 2

[45] Operation [redacted] bundle, DockerImagev3/ 329f1d0f8624640fc204f9aeb4bfec1d203544a1cf32dd38a6b34daebf6043a7 /layer.tar

single exfiltrated message. However in my opinion this is unsound. The evidence supporting (b) includes the following.

a. As the messages are in XML format and the `json_data` file in 'JSON line' (JSONL) format, the intercepted data was reformatted somewhere. This reformatting substantially increased its size. It might have been more logical to do this on a DGSI server rather than on a target handset.

b. The JSONL format may be verbose, but it is ideal for an intelligence agency to maintain a NOSQL database of intercepted material that is aggregated from multiple sources, and from which extracts can then be drawn efficiently with respect to any special selector or other attribute of interest. The JSONL format is often used to aggregate data collected from multiple disparate systems.

c. The JSONL data items include many that are redundant. Why for example include not just the message creation date and the burn time, but also the burn date? That is just the creation data, incremented by the number of seconds of burn time. That may make sense in a database but not in exfil. Furthermore, burn date is computed for some messages but not all.

d. The 30th January Lille warrant says: '*The installation of the technical device for capturing user data and its uninstallation will be carried out through a requisition from the Director-General of Internal Security... using State resources subject to the secrecy of national defence. This service will be responsible for storing the raw data subject to seizure and placing it under seal at the end of the capture period*'[46].

e. Some dates and times in the `json_data` file are stored as Unix Epoch timestamps, whereas others are ISO-formatted strings. It would be bizarre to send datetimes as ISO strings over a network, not to mention repeating those fields multiple times, as the native Unix Epoch datetime encoding is very significantly more compact and faster to process. It would also be bizarre to modify the app to translate machine times into more verbose ones in two different ways. However, if the translation were done on the load balancer or at the DGSI's back-end systems, this might well happen as a result of the reuse of legacy code left over from other interception operations.

f. The Europol minute of the 18-21 February meeting says: '*France should provide the data in a way that the entire conversation between users will be visible even when one is in a different country. Both Netherlands and France agree to this. France will assess if it is technically manageable in the short term*'[47]. The French were also to provide samples of the final format data to participants.

[46] pp 1282 (French) and 1288 (English)

[47] P 1309 no 10

g. An email from 'Nicolas' at the Gendarmerie to Luke Shrimpton[48] states: "…we did a last round of data generation yesterday night on an updated version, here you can find the last sample. We tried to be as extensive as possible (notes, images, change of language, timezone ...). As you guys can see the live data file is now named json data, metadata file being a corrupted one you shouldn't worry about. Also it is now proper aggregation of jsons."

h. In the [redacted] case, Shrimpton considers the correlation between messages intercepted directly from [redacted] and those intercepted by his counterparties [redacted] and [redacted][49]. The latter come from France via an EIO. These raise a number of questions. First, he claims that he could produce a script to generate the CSV files; the resulting Excel spreadsheets have tabs which divide the messages by the geolocation of the correspondent handset and whether it is an X2 or X3[50]. The UK json files do not possess handset model data, or location data for counterparties. This suggests that the underlying French dataset from which the [redacted] data are extracted is a superset of the `json_data` files supplied to the NCA. If so, the lines in the UK files are probably not be a complete record of the data aggregated by the DGSI.

i. Second, he remarks that "the differences identified ... are only partially explained by the implant not being active". Yet as we have seen the attack on the handset was in all probability not an 'implant' but a modified version of the original EncroChat app. Trying to explain the no-data days in this way is unsatisfactory as if the modified app were not working, customers would not be able to use the system.

j. Incoming messages usually appear in the json lines twice. For example, in the received message quoted on pp13–14 of Mr Shrimpton's June 3rd report, the message body appears seven lines from the top on p13 – and again five lines from the bottom on p14. A quick search discloses many other examples in the disclosed data. We discuss this in much more detail below.

k. Some messages in the `json_data` file are present in multiple copies – two, three or even four copies – in some of the dumps.

119. On the matter of legacy code, the server code disclosed to us by the NCA appears to have been written by multiple people. For example, the DNSServer.py code used to redirect the EncroChat traffic contains comments in Spanish as well as English, suggesting that the interception framework was developed over time by multiple agencies in operations against multiple interception targets.

---

[48] R27

[49] Item 473 in [redacted]

[50] Tabs are a custom feature of Excel; I don't know if it's even possible to embed them if a .csv file. But the extra data surely came from somewhere.

Interception of photos

120. My instructing lawyers ask me whether the photos might have been collected directly from the EncroChat server, as in normal chat apps such as Signal and WhatsApp, a thumbnail of each image is sent to the recipient, and when they tap on it the full image is downloaded from the messaging server.

121. EncroChat does indeed appear to use server-based image download. We see the repeated appearance in message bodies of long strings of which the first 20 characters are rO0ABXNyADRuZXR3b3Jr, but with variants further along. On analysis, this is a base64-encoded Java serialization of a Java object that gives a URI to download an encrypted image to some other process. These messages also have the content duplicated if they are incoming messages, and a curious fact is that the first copy of the message comes with a plaintext version of the decryption key, name "key_b64", while the second copy has not just the same decryption key (now named just “key”) but a plaintext download URI. This would be consistent with this download URI being somehow added by the EncroChat message server. The strings starting rO0ABXNyADRuZXR3b3Jr appear to be a historical feature of the Encro system, since we find instances of this prefix in the historical message traffic (e.g. carrothorn, March 28). However the added plaintext versions of the URI and key appear to be an artefact of the interception apparatus.

122. See the following three pages for an image of a sample json taken from the bundle for [redacted], the handle allegedly attributed to the lead claimant, and supposedly received by this device on April 20th at 0921. This single line of json has been formatted by “pretty printing” from the one json line in the file to 99 lines that make clear the structure of the nested data attributes and values within it.

123. This sample illustrates the central problem facing us with the json messages. On the one hand there’s a reference to an Android directory at line 16 which suggests that the json line comes from a phone. On the other, the line itself is an aggregation of the data seen from the perspective of the interception apparatus. A finder of fact who read merely the first few attribute-value pairs might well jump to the conclusion that it was produced by the phone. However I remain unconvinced, for the reasons set out below.

```
{
  "server_time": "1587370895.8476005",
  "metadata": {
    "imei": "358098093388140",
    "imsi": "204080819037658",
    "country code": "gb",
    "ota": "O.Z.218.20190622",
    "creation": 1587370895527,
    "creationTimestamp": 1587370895527,
    "timezone": "Greenwich Mean Time",
    "version": "1.0",
    "path": "metadata",
    "size": 3052
  },
  "filedata": {
    "com.esocrypt.chat.app.im": {
      "message": {
        "id": "58870c65-2ae5-4e58-8108-53864e0a9901",
        "contactId": "371698f0-2bb5-4c4c-9d7b-686f84ddebe3",
        "packetId": null,
        "type": "INCOMING",
        "body":
"
rO0ABXNyADRuZXR3b3JrLmVuY3JvY2hhdC5pbS50cy5jcnlwdG8ucHJvdG9jb2wuTWVkaWFNZ
XNzYWdlBT2eVEr/
vQwCAAZMAANrZXl0ABJMamF2YS9sYW5nL1N0cmluZztMAAdtZXNzYWdlcQB+
AAFMAAhtaW1lVHlwZXEAfgABWwAHcHJldmlld3QAAltCTAAPcHJldmlld01pbWVUeXBlcQB+
AAFMAAN1cmlxAH4AAXhwdABFTUdQZzZ6bklPRGlZK0hTbnJxbnF3UT09OlFyOUVaakg5WmxFc
CsrTEExWXYybjltOU5QcG1WOXlrSnNIMW54Y2NCUDA9dAAAdAAGanBlZy4xcHB0AEA5ZDY5MD
diMzQ1YjAyZjY0NmRjMzE4YTIxMzNhMDBkN2UxYmY4NmVjNjg4ZGYzNzMzMTk1M2U5NjYyMzE
5NDM1",
        "dateCreated": "Mon Apr 20 09:21:34 GMT+01:00 2020",
        "burn": 603153,
        "burnDate": null,
        "status": "RECEIVED",
        "dateRead": null,
        "media": {
          "key_b64":
"
TUdQZzZ6bklPRGlZK0hTbnJxbnF3UT09OlFyOUVaakg5WmxFcCsrTEExWXYybjltOU5QcG1WO
XlrSnNIMW54Y2NCUDA9",
          "image_id": "b79f652b-6d21-4077-85b6-d1fe0f22429a",
          "message_id": "58870c65-2ae5-4e58-8108-53864e0a9901",
          "mime": "jpeg.1"
        },
        "from": {
          "id": "371698f0-2bb5-4c4c-9d7b-686f84ddebe3",
          "status": "Online",
          "nickname": "[redacted]",
          "email": "[redacted]@encrochat.com"
        },
        "to": {
```

```
          "id": "67e15819-6f37-4163-b16e-e85fff158c45",
          "status": "Online",
          "nickname": null,
          "email": "[redacted]"
        },
        "contact_extra": {
          "id": "371698f0-2bb5-4c4c-9d7b-686f84ddebe3",
          "accountId": "67e15819-6f37-4163-b16e-e85fff158c45",
          "username": "[redacted]",
          "domain": "encrochat.com",
          "nickname": "[redacted]",
          "status": "Online",
          "sortFix": false,
          "priority": false,
          "blockedCaller": false,
          "burn": 1295940,
          "identityState": {
            "id": "71f79e12-f9ed-443a-9f7d-7dcc475e8b43",
            "contactId": "371698f0-2bb5-4c4c-9d7b-686f84ddebe3",
            "verificationMethod": "Manual",
            "trustedNotary": true,
            "identityHash": "/i9iCZWlKwIUA0l5znrszu41j/A=",
            "identityVerified": true
          },
          "pendingInvitation": false,
          "invitation": null,
          "lastMessageDate": "Mon Apr 20 09:21:34 GMT+01:00 2020",
          "lastUnreadMessage": {
            "id": "58870c65-2ae5-4e58-8108-53864e0a9901",
            "contactId": "371698f0-2bb5-4c4c-9d7b-686f84ddebe3",
            "packetId": null,
            "type": "INCOMING",
            "body":
"
rO0ABXNyADRuZXR3b3JrLmVuY3JvY2hhdC5pbS50cy5jcnlwdG8ucHJvdG9jb2wuTWVkaWFNZ
XNzYWdlBT2eVEr/
vQwCAAZMAANrZXl0ABJMamF2YS9sYW5nL1N0cmluZztMAAdtZXNzYWdlcQB+
AAFMAAhtaW1lVHlwZXEAfgABWwAHcHJldmlld3QAAltCTAAPcHJldmlld01pbWVUeXBlcQB+
AAFMAAN1cmlxAH4AAXhwdABFTUdQZzZ6bklPRGlZK0hTbnJxbnF3UT09OlFyOUVaakg5WmxFc
CsrTEExWXYybjltOU5QcG1WOXlrSnNIMW54Y2NCUDA9dAAAdAAGanBlZy4xcHB0AEA5ZDY5MD
diMzQ1YjAyZjY0NmRjMzE4YTIxMzNhMDBkN2UxYmY4NmVjNjg4ZGYzNzMzMTk1M2U5NjYyMzE
5NDM1",
            "dateCreated": "Mon Apr 20 09:21:34 GMT+01:00 2020",
            "burn": 603153,
            "burnDate": null,
            "status": "RECEIVED",
            "dateRead": null,
            "media": {
              "id": "b79f652b-6d21-4077-85b6-d1fe0f22429a",
              "messageId": "58870c65-2ae5-4e58-8108-53864e0a9901",
              "mimeType": "jpeg.1",
              "downloadUri":
```

```
83… "9d6907b345b02f646dc318a2133a00d7e1bf86ec688df37331953e9662319435",
84                "storageUri": null,
85                "key":
…  "
…  TUdQZzZ6bklPRGlZK0hTbnJxbnF3UT09OlFyOUVaakg5WmxFcCsrTEExWXYybjltOU5QcG1WO
…  XlrSnNIMW54Y2NCUDA9",
86                "preview": null,
87                "previewMimeType": null
88              }
89            },
90            "lastReceivedMessageDate": "Mon Apr 20 09:21:34 GMT+01:00
…  2020",
91            "localCapabilities": null,
92            "remoteCapabilities": null,
93            "participants": []
94          }
95        }
96      }
97    }
98  }
99
```

124. However, we now understand that the json lines may be aggregated by the DGSI from information collected from multiple sources. This suggests that we might not have to make a binary choice whether all the data were collected from the handset (Campbell's hypothesis) or all from intercept. A third possibility is that the information was collected from both.

125. In the message expanded above, we see a server timestamp followed by five data fields:

a. lines 3–14 are 'metadata' about the target which can be taken from the relevant EncroChat server (e.g. see EIO at p 1352);

b. lines 15–33 appear to refer to the pathname of the EncroChat app in the handsets. They contain at 22 a body consisting of a base 64-encoded serialized Java object containing a download URI and key, and at 29 the first decoded version of the key to decrypt the full image;

c. lines 34–39 have the sender data plus the fact that the sender is online, and unless the recipient knows the sender's online status these records may be from the server;

d. lines 40-45 are the recipient data;

e. lines 46-72 are further data about the sender, including that the sender is not 'pendingInvitation';

f. lines 73–89 may be intercepted. They contain at 73 the same message body, and at 85 the same key, although with a different name. There is now an explicit download URI decoded at 83, showing where the image was

collected. There is however no storage URI as we might expect if the image had been downloaded to the handset and then collected from there;

g. lines 90–93 have the lastReceivedMessageDate, which may be helpful for synchronising the decryption apparatus.

126. One of Luke Shrimpton's analyses of the json data appears in his Venetic Data Source Report of 28th April 2021. In his Figure 1 he starts off by highlighting those data items that in his opinion came from the handset rather than the servers:


```
{"server_time": "1585907030.5144858", "metadata": {"imei": "358098093388140",
"imsi": "204080819037658", "country code": "gb", "ota": "O.Z.218.20190622",
"creation": 1585907029812, "creationTimestamp": 1585907029812, "timezone":
"Greenwich Mean Time", "path": "metadata", "size": 178}, "filedata":
{"android": {"cellLocation": {"type":
"android.telephony.gsm.GsmCellLocation", "info": "[2091,60373,-1]", "country
code": "gb", "network operator": "23430", "network operator name": "EE"}}}}

{"server_time": "1589010728.53402", "metadata": {"imei": "358098093388140",
"imsi": "204080819037658", "country code": "gb", "ota": "O.Z.218.20190622",
"creation": 1589010758341, "creationTimestamp": 1589010758341, "timezone":
"Greenwich Mean Time", "version": "1.0", "path": "metadata", "size": 181},
"filedata": {"android": {"cellLocation": {"type":
"android.telephony.gsm.GsmCellLocation", "info": "[11275,3475713,-1]",
"country code": "gb", "network operator": "23430", "network operator name":
"EE"}}}}
```


Figure 1: Examples of raw cell data collected as part of VENETIC. I have bolded the sections that relate to the collected cell data. The top example comes from Raw/GB_358098093388140_204080819037658_2020_04_03/json_data the bottom example comes from Raw/GB_358098093388140_204080819037658_2020_05_09/json_data

127. His scripts to extract message data from the json files to the csv files used in spreadsheets do not extract the header data that is not in the bold font, which I infer to have been added by the DGSI server.

128. He deduces from his network experiments that copies of incoming messages can be read in json format from the message table in the EncroChat app. However these are present in his experiment once, not duplicated as in the live intercept example we studied above.

129. He deduces from the presence of message read times that messages reported in the json files are read from a device rather than from intercept. This may have been the case before the app was modified, but was not necessarily so afterwards. In any case, the most that one might reasonably infer from this is that some of the data in the relevant json lines were exfiltrated directly from a device, not that all the data were.

130. As for the images themselves, they appear to have been intercepted at the media server. I can see no evidence of their being collected directly from handsets other than in phase 1. Indeed, we discovered the following comment in the NCA Python code for handling images: "Encrypted media folder has filenames which are just the URIs"[51].

[51] The file path for the comment reference is `Code\Realm\decrypt_media.py:61`, where `Code` is the path to which the Python was previously extracted from the Docker image.

131. The tentative conclusion to which I come is that the json lines are a mixture of material intercepted from the servers and material intercepted from the handsets. I use the word 'intercepted' rather than collected in both cases, because the material from the handsets was encrypted using the EncroChat cluster's TLS key, intercepted at the load balancer, and decrypted there. All the material therefore appears to be collected from the Public Telecommunications System when it arrives at the load balancer. I note Burnett LCJ's ruling in R v A B D & C that the boundary of the telecommunications system was defined by encryption. Any argument on this point must of course be made by my instructing legal team.

132. My instructing lawyers ask me whether we understand the mechanics of message receipts, as partly described by Luke Shrimpton in his reports. His report of May 13th shows that the message ID is present in outgoing messages only in the header, not the ciphertext; in the April 28 report he shows that the message ID in the intercept becomes the packet ID in the stored message; and his report of June 3rd shows the message ID appears to be renamed the transport ID during processing, then returned as a received receipt to the sender.

133. I do not agree with a number of the other conclusions in his report. For example he notes a message from carrothorn to [redacted] on April 10th that was intercepted as a received message eight seconds before it was intercepted as a sent message, and concludes that it must have been intercepted at the handsets rather than en route to or from the message server. This does not automatically follow, as the processing paths for senders' and recipients' messages will be different, and the delay could be at a server as well as at a handset.

134. It may be relevant that the duplication of incoming messages does not happen when messages are batched. See for example the batch referred to by Mr Shrimpton at the bottom of p12 of his report of June 3rd (line 17 of the json in the Raw folder for 358098093388140 on May 21). If the batching had been caused by a pipeline stall on the handset, as Mr Shrimpton claims, I can see no explanation for this.

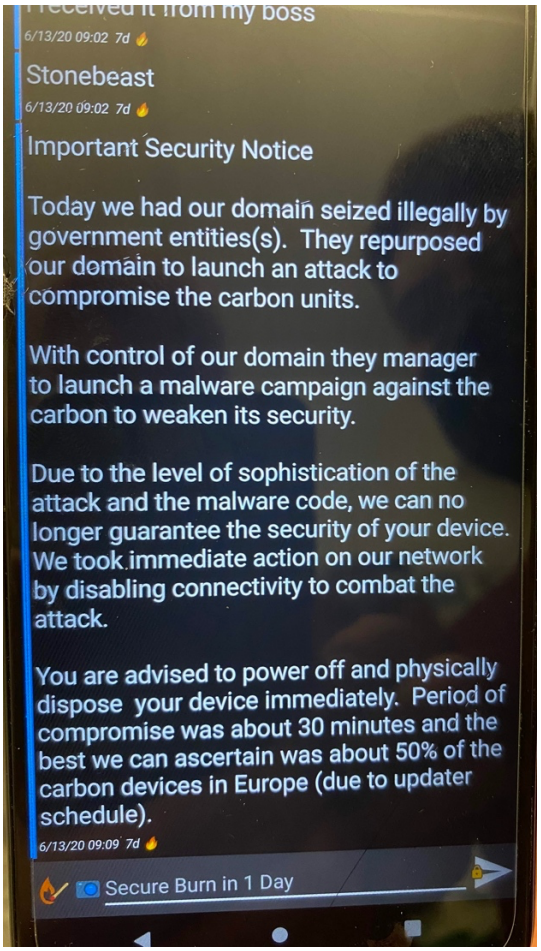
135. My instructing lawyers ask about Shrimpton's experiment, noted in his report of 3rd June, in which he produced the result of using Frida and Jadx to monitor the RAM on a device. If he is suggesting that the PENDING status refers to messages that have been created by the EncroChat app but not yet sent to the server by the Push app, and he also maintains that the modified EncroChat app is the source of all the data in the JSON lines then every one of the intercepted outgoing messages should have the status PENDING. The great majority of them do not.

136. We observed experimentally that EncroChat uses the Axolotl version of Signal which has been obsolete since about 2015, but we have not been able to explore the extent to which even the original EncroChat app implements the Sesame protocol, which is used by Signal to manage message concurrency.

137. Looking through the documents, we also not Jérémy Decou's response of 17/09/2020 (p 1176–85): '*you have to see if there is any data on the location of the base stations and if there is, it is live data*'. This is consistent both with interception from handsets, and with live intercept by message decryption, augmented by handsets

capturing cellsite location when messages are sent or received. He also states that if a message is sent from handset A to handset B, then if the implant is set up on handset A the collection takes place as the encrypted message is sent from the handset to the server; while if it is set up on handset B then the collection takes place once the message has arrived and has been decrypted. This statement makes sense if messages sent from an uninfected phone to an infected one could not be decrypted until a ratchet had been captured for the recipient. However M Decou admits he is an investigator not an engineer.

138. We can see no direct evidence of active attacks or disinformation in the raw data, but I cannot help wondering whether the broadcast message apparently sent by the Encro operator instructing all users to stop using the system and dispose of their handsets was in fact sent by the DGSI / JIT impersonating the operators. The DGSI / JIT access to the key server would have enabled impersonation of any system user. The disruptive takedown of EncroChat was one of the Op Emma declared goals[52]. I reproduce a picture of this message below.



139. I would like to record that this work had to be completed under time pressure. We got the working Carbon handset on December 17th, got the network experiment working on December 23rd, and Mr Boucher had to leave for the US the following

[52] P 1272

day to spend Christmas with his family. Although we had asked for a deadline of January 10th the court asked me to finish this report by January 4th.

140. In conclusion, it is my professional opinion that the interception operation run by DGSI and the JIT was much more complex than the courts have considered so far.

a. The EncroChat telecommunications system consisted of the EncroChat app, running on at least three different types of phone, plus a central infrastructure of 70-plus virtual servers at OVH.

b. Interception involved unauthorised modification of the app and the replication of this infrastructure, with the addition of an active *'mesure de captation'* in the form of a load balancer to switch traffic between the genuine EncroChat servers and copies of them introduced for the purpose of interception.

c. All the data – even the phase 1 sweeps that were copied from handset Realm storage – were encrypted at the handset and then collected at the central server infrastructure of this communications system which had been modified in this way. They were thus intercepted from the telecommunications system if we follow Burnett LCJ's ruling in A B D & C.

d. The intercepted material was collected through a load balancer operated by DGSI, the French internal security service, and aggregated on servers under its control. The images were most probably intercepted from the media server rather than from the handsets, and the first few data records of each json line were most probably written by the DGSI server. Some of the message contents were probably obtained at least in part by decryption. Although we have not been able to exclude the possibility that they were all collected directly from the handsets, that looks unlikely as the data look quite unlike the pre-implant EncroChat messages reported by the NCA. Even if the modified app did upload substantially all the non-image message content directly, then this exfiltrated material was still encrypted in the handset using the EncroChat TLS key and intercepted by DGSI using its load balancer at the EncroChat server cluster in Roubaix.

e. To my understanding this satisfies the definition of interception in section 4 of the Investigatory Powers Act. As I am not a lawyer, I leave the detailed argument to the lawyers appearing before the Tribunal.

f. My professional experience does however cover how policymakers and engineers understand the term bulk interception, and this is a classic case of it. If the Chinese were to perform a similar attack on the Signal messaging app, it would be described as bulk interception. If the Tribunal were to take a different view in this case, that would open a real gap between how engineers describe interception and how UK lawyers would subsequently have to.

141. We have not managed to identify the channel on which the location messages were leaked by the modified app. We have also not managed to establish how much of the JSON line data, and indeed how much of the other evidence offered, came from the handsets directly and how much from the servers. If the Tribunal upholds the

warrant, then experts in the crown courts will be called on work out how much of the presented data can be relied on.

142. If the NCA wishes to maintain that the JSON lines of message metadata and contents were all leaked directly from the handsets, then they should supply (or the court should order the discovery of) the following:

a. The source code for the modified app;

b. The apk files for the modified app;

c. Information required to set up a network experiment to capture the location messages from a phone with the modified app;

d. A working X2 phone with the modified app and the passwords required to unlock it;

e. Copies of any other software used to generate the json line files in the evidence bundles.

I understand that it is my duty to help the court understand matters within my expertise and that this duty overrides any obligation from the person from whom I have received instructions. I have complied with that duty and will continue to do so. I have done my best to be accurate and complete.

I confirm that I have made clear which facts and matters referred to in this report are within my own knowledge and which are not. Those that are within my own knowledge I confirm to be true. The opinions I have expressed represent my true and complete professional opinions on the matters to which they refer.

I understand that proceedings for contempt of court may be brought against anyone who makes, or causes to be made, a false statement in a document verified by a statement of truth without an honest belief in its truth.

Signed



Professor Ross Anderson FRS FREng

Cambridge, 5th January 2022