UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

MATTHEW CAMPBELL and MICHAEL HURLEY, on behalf of themselves and all others similarly situated,

Plaintiffs

v.

FACEBOOK, INC.,

Defendant

Case No. C 13-05996 PJH

Expert Report of Dr. Benjamin Goldberg

Table of Contents

I. QUALIFICATIONS 1

II. SUBJECT MATTER OF OPINIONS 2

III. MATERIALS REVIEWED 2

IV. SUMMARY OF OPINIONS 3

V. TECHNOLOGY OVERVIEW 8

VI. THE FACEBOOK SOCIAL NETWORKING SERVICE 10

VII. THE FACEBOOK SOFWARE/SOURCE CODE 16

A. URL Previews and Attachments 16

B. [redacted] and Nectar Logging 24

VIII. RESPONSE TO THE GOLBECK REPORT 26

A. The Alleged "Interceptions" 26

1. Creation of Share Objects and Incrementing Global Objects 27

2. [redacted] and Nectar Logging 30

3. Share Scraper/Web Crawler 31

4. Code-Based "Devices" 31

B. Variability in the Purported "Interceptions" 32

C. Variability in the Alleged "Uses" of "Intercepted" Data 37

D. Continuing Conduct 41

E. Lack of Ascertainability of Class Members 41

F. Variability in Operation of the Source Code Across Senders and Recipients 44

IX. COMPUTER STORAGE 44

A. Computer Storage Technology 44

B. The Incorrect Distinction Between "Memory" and "Storage" 47

X. ORDINARY COURSE OF BUSINESS 49

XI. RESERVATION OF RIGHTS 51

I. QUALIFICATIONS

1. I am a tenured Associate Professor in the Department of Computer Science of the Courant Institute of Mathematical Sciences, New York University ("NYU"), in New York, NY. I have held this position since September 1994. From 1987 to 1994, I was an Assistant Professor in the Department of Computer Science at NYU. Since September 2014, I have been the Director of Graduate Studies for the MS programs in the Department of Computer Science, having previously served in that role from September 2009 through August 2012. I served as the Director of Undergraduate Studies for the Department of Computer Science from September 1995 through August 1998 and from September 2003 through August 2006. In addition, I have held a one-year visiting professorship at the Institut National de Recherche en Informatique et en Automatique (INRIA), a national laboratory in France, and was twice appointed to a month-long position as an invited professor at the Ecole Normale Supérieure, a university in Paris.

2. I received my Doctoral degree in Computer Science from Yale University, New Haven, Connecticut in 1988, having previously received Master of Science and Master of Philosophy degrees in Computer Science from Yale in 1984. My undergraduate degree from Williams College in 1982 was a Bachelor of Arts degree with highest honors in Mathematical Sciences.

3. I have taught courses at the undergraduate and graduate level in, among other things, software development, programming languages, object oriented programming, hardware design, and operating systems (including networking). I have also published extensively in the areas of memory management and distributed and parallel computing, which involves computers connected by networks and data transmission between computers.

4. I have testified numerous times as an expert in computer science at trial in U.S. District Court, including in the U.S. District Court for the Northern District of California. I have

also provided two technology tutorials to the Court in the U.S. District Court for the Northern District of California. Additional information concerning the computer science courses that I have taught, my professional publications and presentations in the field of computer science, and cases in which I have testified as an expert at trial or deposition in the past four years are set forth in my current Curriculum Vitae, a copy of which is attached as Exhibit BBB.

II. SUBJECT MATTER OF OPINIONS

5. I have been retained by Gibson Dunn, counsel for Facebook, to form opinions regarding the operation of the Facebook social networking service, to explain the meaning of computing terminology related to this matter, and to respond to certain assertions made by Dr. Jennifer Golbeck in her November 13, 2015 Report in Support of Plaintiff's Motion for Class Certification ("the Golbeck report"). (Dkt. No. 137-6.)

6. I am being compensated at a rate of $450 per hour for my work in this matter. My compensation is in no way conditioned on the nature of my opinions or on the outcome of this matter.

III. MATERIALS REVIEWED

7. A complete list of the materials that I reviewed in forming my opinions set forth in this report is attached hereto as Exhibit CCC. As detailed further below, I also reviewed portions of Facebook's source code from the period of September 2009 to December 2012, which I understand to be the same source code to which Dr. Golbeck had access. Consequently, unless otherwise specified, my opinions are limited to the operation of Facebook's messaging service during that period.

IV. SUMMARY OF OPINIONS

8. I disagree with Dr. Golbeck's conclusions that Facebook's implementation of what I or any other skilled computer scientist would consider routine programming functions are "interceptions" of message content using what she calls "code-based devices" (a term I have never heard before). Dr. Golbeck concludes that Facebook's creation of an "EntShare" object, representing a URL attachment to a message, is an "interception," but she does not draw the same conclusion about any of the other objects that are generated when a message is sent. Creation of objects like EntShares is not unusual in object-oriented programming, and here, Facebook actually uses EntShare objects to display the URL attachment to the recipient and sender of the message when they view it in their inbox.

9. Dr. Golbeck also claims that Facebook's logging of some share events in Nectar, and storage of some share data and counters in the "EntGlobalShare" and in the [redacted] Hive table also constitute interceptions. However, the [redacted] Hive table was deleted prior to the Class Period, and, in any event logging events and storing activity data are processes performed by nearly all software systems to track error rates, resource usage or congestion, and security concerns, among other things.

10. Even if Dr. Golbeck's characterization of these standard programming practices as "interceptions" were credited, an individual, message-by-message inquiry would be required to determine whether a given URL in a given message was actually "intercepted." For instance, there were many circumstances in which URL previews were not generated, in which case neither a URL attachment nor an EntShare object could have been created. And even if a URL preview and attachment *were* generated, the sender could have deleted the URL attachment before sending the message, or deleted the URL in the message before sending it. Alternatively, Facebook's code for security functions could have prevented generation of a URL preview, or

later, generation of an Entshare object, for a URL deemed malicious based on Facebook's internal "blacklist" of URLs. Further, even if an EntShare object was created, the counters in the EntGlobalShare might not have been incremented. This is even more true for logging share scrapes to Nectar, which only logged a sample of share scrapes (and, in any event, logged the share scrape before the message was even sent).

11. Similarly, to determine whether URLs shared in messages were "used" in the ways Dr. Golbeck discusses would also involve a message-by-message inquiry (and in several instances, might not be possible at all). For instance, Dr. Golbeck discusses Facebook's alleged inclusion of URL shares from messages in: the aggregate demographic data reflected in Facebook's "Insights" feature (an interface for website owners to view data about engagement with their sites); aggregate, anonymous Plugin Counters (numbers that website owners can display with Facebook's social plugins that reflect the level of engagement with those sites); aggregate, anonymous counters in the "link_stats" table and Graph API (publicly available counts that reflect the level of engagement with a given URL); and Recommendations and Activity Feeds (which displayed a list of pages for a given website that the viewer may be interested in seeing). However, determining whether a given instance of sharing a particular URL in a message was actually "used" for any of these purposes is likely not possible and, at a minimum, would depend, among other things, on:

i. The date the message was sent, since each "use" was only in operation for, at most, a finite period of time within the proposed class period;

ii. Whether a URL preview was successfully generated;

iii. Whether an EntShare object was successfully generated;

iv. Whether the counters that supply these features were actually incremented for a given instance of sharing the URL in a message; and

v. Whether the relevant features were actually accessed or implemented, including whether anyone actually accessed the interfaces or APIs to view that aggregated share data for that URL, or whether the website for that URL actually included the plugin that included or considered aggregate shares of the URL.

12. Dr. Golbeck cannot validly opine on the current operation of the messages product since she, like I, could have examined, and at most did examine, only Facebook source code operational until December 31, 2012 (per the agreement of the parties). Accordingly, while her report expresses conclusions about the *current* operation of Facebook software, I do not believe she has a factual basis for these conclusions since there is no way for her to have reliably determined the behavior of Facebook's software for any date after December 31, 2012. Her attempt to interpret internal Facebook emails allegedly bearing on potential practices (most or all of which appear to be written long before the beginning of the Class Period) does not form a valid basis for these conclusions about the operation of the current source code (or source code from any time period).

13. Dr. Golbeck also provides "sample" code for a database query of her own design, which she claims would yield a list of the Facebook IDs of everyone whose actions had created an EntShare from a Facebook message, and she has stated that this would identify the members of Plaintiffs' proposed class. However, her proposed query is both under- and over-inclusive in a number of ways (all of which I understand she has conceded). First, it will be under-inclusive in that it will not identify:

i. Recipients of messages with URL attachments;

ii. Senders and recipients whose messages with URL attachments were deleted;

iii. Senders and recipients whose accounts were deleted;

iv. Senders whose messages were blocked for site integrity purposes;

v. Senders whose URL attachment did not result in the creation of an EntShare object for any reason; or

vi. Senders that deleted a URL attachment before it was sent.

Dr. Golbeck's proposed query will also be over-inclusive in that it will include:

vii. Senders whose messages did not contain URLs in their text;

viii. Senders who never typed a URL into a message, and instead merely chose to "Share" a URL through a "Share" button on a third-party website;

ix. Senders and recipients outside the United States;

x. Senders of messages outside the Class Period; and

xi. Senders that were not subject to the challenged "uses."

Her proposed query also cannot identify senders:

xii. Subject to Nectar logging;

xiii. Whose shares incremented an EntGlobalShare, the "link_stats" table or Graph API, or the plugin counts on a third-party website; or

xiv. Whose share data was considered or displayed in any Recommendations or Activity Feed plugin, or was displayed or viewed in Insights data or any API available to third parties.

14. As discussed below in further detail, Facebook's source code and software did not operate the same way for all purported class members. There was substantial inherent variability based on things like the evolution of the features and their software over time and implementation issues (race conditions, data corruption, database failures, etc.) inherent in any system as large as Facebook's.[1]

15. I disagree with Dr. Golbeck's distinction between "memory" and "storage," in which she claims that, when a Facebook message and URL attachment are processed on

[1] Race conditions occur when too many calls are made and not all of the share objects are properly counted by the internal counter. Database failures and contention may result in failure to properly count all share objects, due to databases being offline or not functional for other reasons, or for databases to fail to agree on a given increment.

Facebook's servers, they are held in memory, not storage. I have never encountered such a distinction in my decades of experience as a computer scientist. Computer systems use different forms of electronic storage, including but not limited to CPU storage, CPU cache memories, random access memory (or RAM), and disk or flash memory; indeed, memory in computing is a form of storage. But even under Dr. Golbeck's unique definition of storage (with which I do not agree), the message data is in storage when it is processed by Facebook's servers, as explained further below.

16. Finally, I disagree with Dr. Golbeck's conclusion that Facebook's alleged interception, analysis and use of URL shares from messages was not necessary for the messaging functionality on Facebook. First, while I do not offer an opinion on illegality as a legal conclusion, I note that Dr. Golbeck appears to use an incorrect inquiry for determining whether any interception was performed in the ordinary course of Facebook's business. Dr. Golbeck assessed whether the alleged interception was "necessary." However, I understand the Court has instead stated that the relevant inquiry is whether there exists any nexus between the alleged interception and Facebook's ultimate business, which is to allow people to share information and connect with other people in a safe, efficient online space. Since my review of the source code and reliance on sworn testimony of Facebook's engineers indicate that Facebook's EntShare objects were used (i) to render URL attachments for senders and recipients and (ii) for site integrity and anti-abuse efforts; and Facebook's EntGlobalShare objects and their counts were used (a) to create URL previews, (b) create programming and operational efficiencies, and (c) for site integrity and anti-abuse efforts, I conclude that EntShare and EntGlobalShare objects are both necessary and also have a clear nexus to Facebook's messaging service *and* its ultimate business.

V. TECHNOLOGY OVERVIEW

17. The technology at issue in this matter relates to the messaging feature of the Facebook social networking platform. The specifics of the Facebook platform are described below in a subsequent section of this report. This section provides a general overview of the technology and defines various technical terms used throughout my report.

18. Facebook, like most services found on the Internet, utilizes a client-server architecture. In a client-server architecture, people on "client" devices, usually personal computers and mobile devices such as smart phones and tablets, interact with a user interface supported by software—generally referred to as "client software"—running on their client devices. More demanding computations, storage of larger amounts of data, and communication between different people are supported by server computers, which are larger, more powerful computers with large amounts of storage capacity. The software that the server computers run, generally referred to as "server software", usually responds to requests from numerous client devices, generates the data to be provided to the clients (often specifying the format of the display of the data), interacts with database management systems to store and retrieve data, and, in many cases, communicates with other server computers. The terms "client" and "server" are used to refer either to the client and server software or to the client and server machines—or both. In this report, unless stated otherwise, I will use "client" and "server" to refer to the software and hardware together. For example, when I discuss a "Facebook server", I am speaking of Facebook server software running on a Facebook server computer.

19. Clients and servers in many modern client-server systems use HTTP, the Hypertext Transport Protocol defined by the World Wide Web consortium (*see* http://www.w3.org/Consortium/) to communicate with each other, almost always over the Internet. Services provided by servers to clients using the HTTP protocol are usually referred to

as "web-based" services. Clients request resources, such as documents (e.g. web pages, audio files, and videos) and services (for shopping, travel, etc.), by issuing HTTP requests. An HTTP request is directed to the proper server—and specifies the desired document or service on that server—using a uniform resource locator (URL). Examples of URLs include https://www.google.com/, http://www.ford.com/new-cars/, and https://en.wikipedia.org/wiki/Webserver_directory_index.

20. People on client devices can communicate with each other in numerous different ways, such as via electronic mail (email), video chat, texting, and messaging. In each of these media, the data being transmitted is generally sent from a client to a server, which then may forward the data to another server, and so on, until the data is sent from a final server to the receiving client (of course, in some instances, there may only be a single server involved). When the data arrives at a server, it is stored on that server at least temporarily until it is forwarded to the next server or to the receiving client. In many cases, a server only forwards the data to the receiving client when the receiving client requests the data. For example, in many email and messaging systems, the email or message data is stored in the recipient's "inbox" on a server until the email or messaging software on the client requests the data. The process of storing transmitted data on a server and forwarding the data to the next server or to a client is known as "store-and-forward".

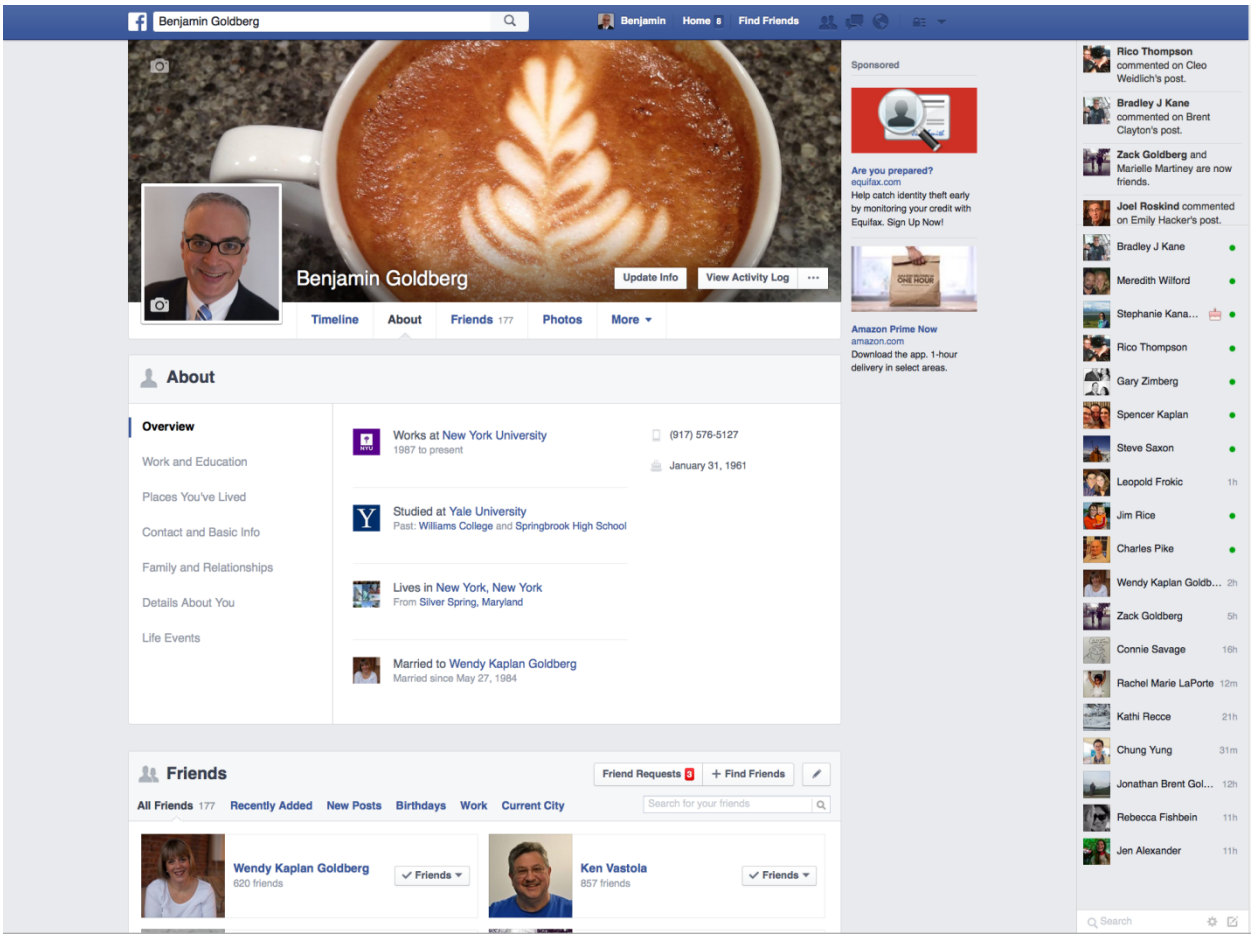
21. In most email systems and some messaging systems, the text of a message that someone has typed can be augmented by an "attachment." An attachment is additional data that is transported to the recipient along with the main text of the email or message, but is often not visible within the display of the text of the email or message. Attachments are often in a different format from the text of the email or message, such as when the text is in plain text or

HTML format and the attachment is a document in PDF format, an image, or an audio or video file. In some cases, the attachment only becomes visible (or audible) when the sender or recipient clicks or taps on an icon or link displayed along with the message. In other cases—as in the Facebook messaging feature described below—the attachment may be displayed automatically when the text of the message is displayed.

22. In most web-based email and messaging systems, even when an email or message is forwarded to the client for display to the sender or recipient, the email or message (along with any attachment) remains in the "inbox" on the server. This way, emails or messages can be accessed by the senders and recipients at a later time or from a different client device and does not require a substantial amount of storage on the client devices. In such cases, the email or messaging service is said to be "hosted" on the server(s) supporting the service, because the server(s) provides the long-term storage for the emails and messages.

VI. THE FACEBOOK SOCIAL NETWORKING SERVICE

23. Facebook is a free social networking service. The main purpose of the service, as stated in Facebook's mission statement, is to give people the power to share and make the world "more open and connected." When people join the service, they create a profile by providing Facebook with a variety of information such as Name, Age, Profile Picture, City, Education, and Work experience. As an example, below is an image of my profile on Facebook.



24. As individuals continue to use the service, they may post additional information that they wish to share. Some of this information may be user-generated, such as photos, videos or posts, and some of it may be information from other sources, such as web articles that someone finds interesting. People who use Facebook can also define other people on Facebook as "friends," and they can control who else on and off Facebook can see what they post (such as "friends," "friends of friends," the general public, and so on). These privacy settings can be configured individually by each person who uses Facebook.

25. In addition to posting and viewing information, people who use Facebook can interact with each other and with information in many different ways, for example:

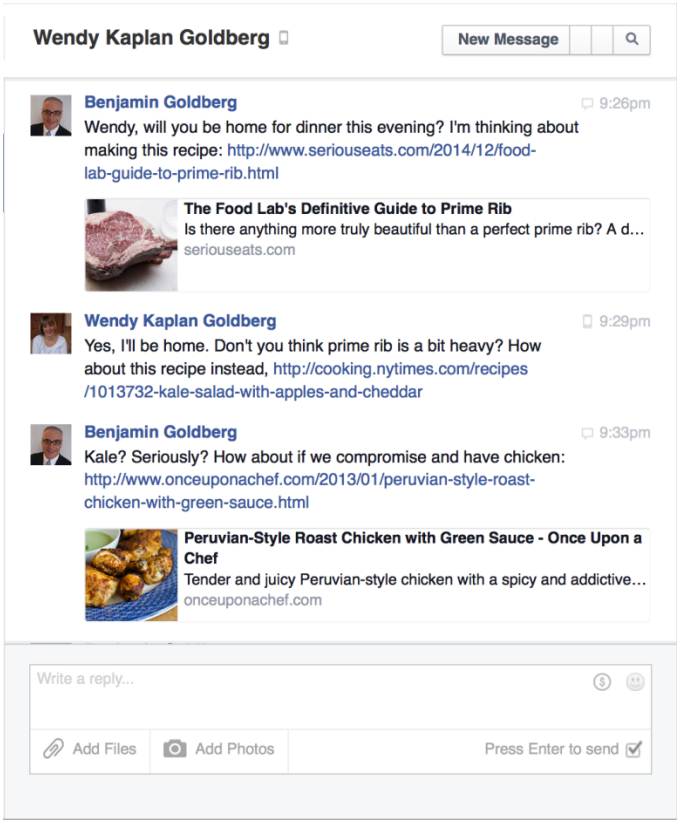
- Pages: Companies or businesses can create pages on Facebook, which enables them to communicate information to people who follow them.
- Groups: People who use Facebook can create groups on Facebook around specific interests or affiliations, which enables communication between members of the group, whether they are Facebook friends or not.
- Applications: Third-party developers can create applications that run on the Facebook platform; many of the most popular applications are games that people play with their friends on Facebook.

26. People who use Facebook to share what they are interested in can also use social plugins, which appear on third-party websites, to share these interests. Plugins like the "Share," "Like," and "Send" buttons let people seamlessly tell friends what has caught their attention, what is important to them, or what they enjoy, even when they are on third-party sites, so they can engage and connect with others on those topics, and so that they can receive other information tailored to their interests and preferences. This helps bring together what is otherwise a sprawling, disjointed network of activity across the Internet and allows people who use Facebook to have a richer, more meaningful connection to other people, ideas, and information. When someone is logged in to Facebook, Facebook can display personalized streams of information from her friends, such as life events, reviews, and comments, anywhere a site has incorporated Facebook plugins. Even when people who use Facebook are not logged in to Facebook, social plugins allow website owners to present information about aggregate engagement with different kinds of information, which can help people identify information that is popular and that they might find interesting. A list of the current Facebook social plugins can be found at https://developers.facebook.com/docs/plugins.

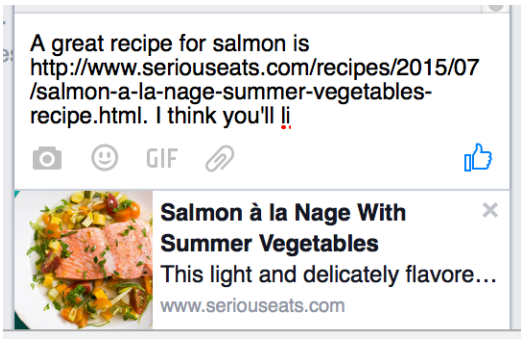
27. While most content on Facebook is a form of "one-to-many" communication, Facebook also provides means for "one-to-one" or "one-to-few" communication. Initially, this was performed through a function called "inbox" which behaved similarly to e-mail, enabling people to send messages to other people directly. This changed over time, and ultimately evolved into "Messages," which is Facebook's integrated communications platform. It enables people who use Facebook to send messages directly to other people, or to groups of people. This platform can currently be accessed through the Facebook website or through its standalone messaging application (Messenger). In addition to sending text, people can send documents and pictures, among other things. If the recipient of the message is currently browsing the Facebook website, the message will appear onscreen and enable her to respond, which can lead to a conversation with messages being sent and received in real time. If the recipient is not online, the message will appear the next time the recipient accesses Facebook. This can lead to asynchronous communication similar to e-mail, where messages are sent and received by the client device at different times. Messages are stored on Facebook servers, which enables people who use Facebook to conduct conversations seamlessly on multiple devices, for example, by beginning a conversation on a PC, and continuing it on a smartphone. It also enables senders and recipients to view messages they have sent or received in the past.

28. For a message sent from the Facebook website that contains a URL, Facebook attempts to enhance the user's experience of the message by generating an attachment to the message. For example, if a sender types a URL for a webpage into a draft message, Facebook will sometimes generate a small preview of the webpage (referred to in this report as a "URL preview"), including both an image and some text from the web page. This process is done before a sender sends the message, giving the sender the option to remove the attachment if the

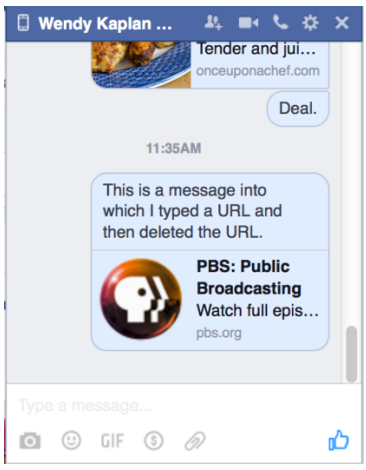
sender wishes. Below is an image of a messaging conversation that I had with my wife (though this one was created for the purposes of this report). Two of the shown messages contain URL previews, and one does not.

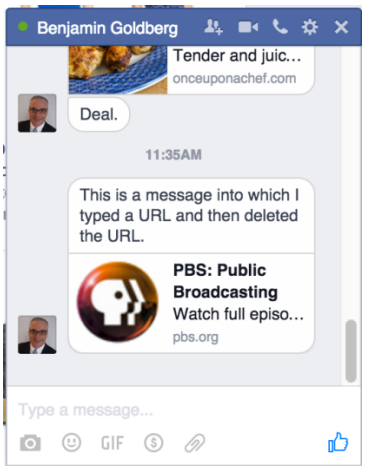


29. As mentioned above, a URL preview is displayed and the corresponding URL attachment is attached to the message if the Facebook software recognizes a URL before the sender sends the message. However, if the sender deletes the URL preview by clicking on an "x" in the corner of the displayed preview while the message is being composed, no URL attachment will be sent with the message. The URL preview display for a message being drafted, along with the "x" to click on to remove the preview, is shown below.



30. Similarly, if the sender decides to delete the text of the URL from the draft message, the URL preview and attachment remain (unless the sender deletes it). Below is a message to my wife in which I originally typed a URL, but then deleted the URL. The URL preview for that deleted URL remains, as does the URL attachment that gets sent with the message. Shown on the left is how the message, after being sent, appears in my inbox and on the right is the message as it appears in my wife's inbox.





This example shows that the URL attachment gets sent with the message, even though the URL is not in the message itself.

VII. THE FACEBOOK SOFTWARE/SOURCE CODE

A. URL Previews and Attachments

31. The Facebook source code that I examined in this matter, which was also available to Dr. Golbeck and was cited in her report, included code from September 2009 to December 2012 and was primarily written in two programming languages: the Facebook messaging client software at issue in this matter is written in JavaScript, and the Facebook server messaging software at issue in this matter is written in PHP. Both of these languages provide support for what is known to programmers as "object-oriented programming." Although a detailed discussion of object-oriented programming is beyond the scope of this report, a core feature of all object-oriented languages is the creation of "objects," which are data structures that both contain the data and specify the operations that can be applied to the data. A data structure is simply a way of aggregating distinct pieces of data so that they can be treated as a unit. For example, a data structure representing a personnel record may include fields for an employee's name, social security number, department, office address, phone, and salary. If that data structure were created in an object-oriented language, the data structure would be called an "object" and, in addition, might specify the operations that can be performed on a personnel record, such as "contact the employee," "adjust the salary of the employee," "change the employee's department," etc.

32. I describe here the functionality of Facebook messaging that is relevant to this matter, based on the source code that I examined and other documents provided to me (including deposition testimony, witness declarations, interrogatory responses, etc.). However, as discussed in subsequent sections of this report, I note that Facebook is a huge system comprising tens of millions of lines of code, hundreds of thousands of servers supporting over 1.5 billion people, and handling an average of over 1 trillion requests each day. (*See* January 14, 2016 Declaration

of Alex Himel ("January 14 Himel Decl.") ¶ 28 n.3.) In such a huge system with that level of demand, there are bound to be software errors that lead to inconsistencies in the behavior of the system and occasional server and database failures that affect the operation of the system. Thus, my description below of the functionality of the Facebook code, particularly the code supporting Facebook messaging, reflects how the system was designed to operate generally—not necessarily how it actually operated at any given moment.[2]

33. Beginning in or around August 2010, Facebook implemented a "share scraping" functionality in messages composed on the Facebook website. (*Id.* at Ex. C (June 1, 2015 Declaration of Alex Himel ("June Himel Decl.")) ¶ 18, Ex. D.) After that date, if someone was typing a message on the Facebook website, accessed using a JavaScript-enabled browser, then the Facebook JavaScript code running on the client device browser would have attempted to recognize a URL within the text being typed (the process of recognizing different parts of a string of text is called "parsing"). There were instances when the JavaScript code may not have recognized a URL, such as if the sender sent the message quickly or if the sender did not hit the space bar after typing the URL. (*Id.* ¶ 20.) If the JavaScript code recognized a URL, it sent an HTTP request to a Facebook server specifying the URL that the sender had typed. [redacted] [redacted] (FB000027089-27110) at lines 800-851; [redacted] (FB000027185-27189) at lines 153-210; [redacted] (FB000027135-27136) at lines 30-32; *see also* FB000027054; FB000027055.) The Facebook server then determined if a preview image of the web page specified by the URL, along with a small amount of text extracted from the web page, had already been stored on a Facebook server. (*See* S[redacted] (FB000027163-

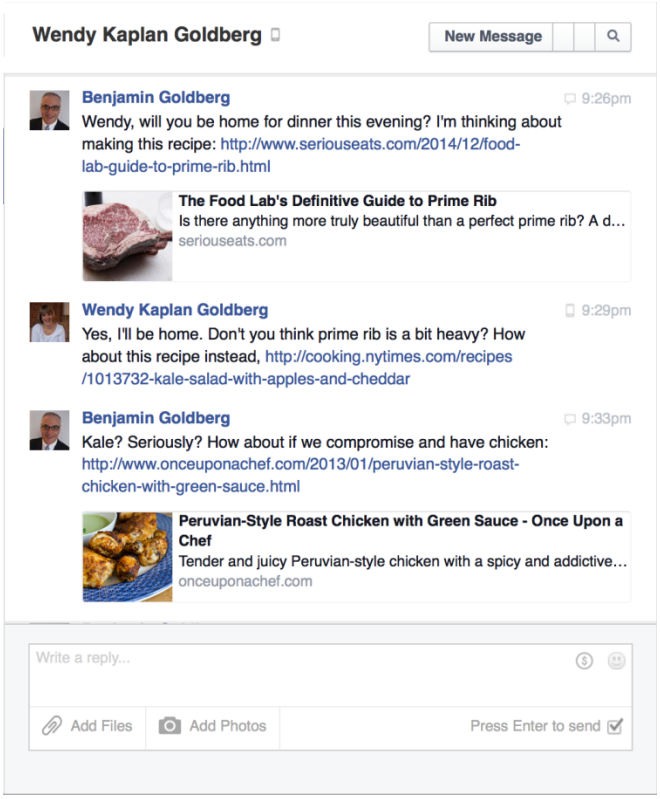
[2] I understand that certain Facebook engineers have submitted declarations in connection with this matter that explain and discuss variation among the practices and functionality challenged by Plaintiffs. Much of that variation is also discussed below.

27170) at lines 107-126; [redacted] (FB000027181-27184) at lines 1-172.) If no stored image and text for the web page was available on the server, the server issued an HTTP request for the specified URL to retrieve the web page and, upon receiving the web page, created a preview image of it and extracted some text from the page. (*See* [redacted] (FB000027149-27162), lines 208-280.) This act of retrieving a web page in order to generate a preview image and extract text from it is known as a "scrape" (a term used generally to mean extracting data from a web page). In Facebook terminology, since the scrape was used to create a preview image and extract text from the web page specified by a URL being shared in a post or message, it was known as a "share scrape." (*See* January 14 Himel Decl. ¶ 14.) Web scraping is a normal technique used to extract data from websites for various reasons. This share scrape process for generating URL previews is also discussed in the Declaration of Alex Himel being submitted simultaneously in support of Facebook's Opposition to Plaintiffs' Motion for Certification. (*See id*. ¶¶ 13-15; *id*. at Ex. B (Facebook's Supplemental Responses and Objections to Plaintiffs' First Set of Interrogatories ("Supp. Resp. to First Interrog.")) at 17-18.)

34. On the Facebook server, when the share scrape had been performed for the first time for a specified URL, an object (again, simply a data structure) was created to record the fact that a URL preview comprising a preview image and associated text had been created so that, for subsequent references to the same URL in Facebook posts or messages being typed, the same URL preview could be used. This object, sometimes called a "global share object" and referenced in the code as an EntGlobalShare, contained information about the URL including, among other things, the URL preview. Additionally, the EntGlobalShare object contained fields (among numerous others) that represented various counts, including a count of the number of Facebook posts that attached the URL (the "post_count" field) and a count of the number of

messages that attached the URL (the "share_count" field). Since the share scrape and the creation of the EntGlobalShare object occurred while the post or message was being drafted by the sender—i.e. *before* the post was actually posted or the message was sent—the "share_count" and "post_count" fields in the EntGlobalShare object would initially have been zero. (*See* [redacted] (FB000027171-27180) at lines 12, 236-274, 314-354.) As discussed below, the "share_count" field of an EntGlobalShare object could not have been incremented unless the message with corresponding URL attachment was subsequently sent. (*See* January14 Himel Decl. ¶¶ 24, 27.)

35. After creating a preview image and description of the scraped web page and creating the corresponding EntGlobalShare object, the server sent the preview image and description back to the client, where the JavaScript code displayed the preview image and description beneath the post or message being drafted. (*See* [redacted] (FB000027089-27110) at lines 853-862.) Below is an example of preview images being displayed after the message was sent (*see also* paragraph 29 for an image of the preview being displayed before being sent).



36. If an EntGlobalShare object for a URL already existed when the server received a request from client JavaScript code for that URL, then no share scrape needed to have been performed. The server simply retrieved the URL preview based on information in the EntGlobalShare object and sent the URL preview back to the client software for display. Again, this process ocurred *before* the post or message that the sender was typing had been posted or sent, respectively. (*See* [redacted] (FB000027135-27136) at lines 30-32; [redacted] (FB000027163-27170) at lines 107-126; [redacted] (FB000027181-27184) at lines 1-172; *see also* January 14 Himel Decl. ¶¶ 13-14; Supp. Resp. to First Interrog. at 17.)

37. I understand from Facebook's interrogatory responses that there was variability in the types of previews that may have been rendered (if indeed a preview was rendered). For

example, if the URL a sender wanted to send required a sender or recipient to log into the destination website, the preview may have been blank, the sender may have received an "HTTP 404" or "Not Found" error message, or the preview may have shown the default page for the website. (*See* Supp. Resp. to First Interrog. at 17.)

38. The use of EntGlobalShares has many benefits. For example, because they eliminate the need to send repeated HTTP requests to the website to fetch the URL preview information, they greatly reduce excess traffic across the Internet and avoid causing crashes or otherwise slowing performance on those websites. They likewise give people a consistent preview experience. Finally, the use of EntGlobalShares is able to provide URL previews to people even when the website indicated by the URL is off-line, having technical difficulties, or is otherwise unable to provide the specified web page. (*See* January 14 Himel Decl. ¶ 14.)

39. In the case of Facebook messaging, when the JavaScript code received the URL preview from the server, the code appended an attachment, referred to as a "URL attachment," to the message representing the preview. The sender had the option to delete the attachment before the message was sent. If the sender did not delete the attachment, however, the attachment would have been sent along with the message once the message was sent. It is important to note, however, that (as discussed above in paragraph 30) if the sender deleted the URL from the text of the message, the URL attachment would have remained attached to the message when the message was sent (unless the sender separately deleted the attachment)—even though the URL itself was no longer part of the message text field. This is an illustration of the fact that the URL attachment was distinct from the message itself. (*See* January 14 Himel Decl. ¶¶ 15-16; Supp. Resp. to First Interrog. at 18.)

40. When the sender had completed typing the message, she could send the message by clicking on or tapping a "Send" button or hitting the enter key. At that point, the message would have been sent—along with the URL attachment—in an HTTP request to the server for subsequent delivery to the recipient. (*See* M[redacted](FB000027089-27110) at lines 545-554.) A Facebook server, upon receiving the message from the client, would have processed the message in numerous ways, including (1) ensuring that the content and attachment were not malicious or junk ("spam"), (2) collecting data about the message and its attachment, and (3) storing the message and attachment in the inboxes of the sender and recipient so that they could view the message text as well as the URL attachment. (January 14 Himel Decl. ¶¶ 16-18; Supp. Resp. to First Interrog. at 18-19; January 14, 2016 Declaration of Michael Adkins ("Adkins Decl.") ¶¶ 10-14.) The server code processing the message was written using the PHP language in which—as discussed above—a fundamental data structure is the "object" and objects are created in the normal course of execution of PHP programs (and not just Facebook PHP programs). Thus, this processing of the message on the Facebook server would likely have resulted in the creation of several new objects. One such object, sometimes called a share object (referred to in the code as an EntShare), was the data structure used to represent the URL attachment to the message. Other objects that would likely have been created (though not based on the URL attachment) included "EntMessage" and "EntMessageThread" objects, the latter of which represent message threads. (*See* January 14 Himel Decl. at Ex. A (Facebook's Second Supplemental Responses and Objections to Plaintiffs' Narrowed Second Set of Interrogatories ("Sec. Supp. Resp. to Sec. Interrog.")) at 18.)

41. The creation of EntShare objects was part of the message delivery process for messages containing URL attachments. The EntShare object corresponding to a URL

attachment to a Facebook message was used in rendering (displaying) the URL preview represented by the URL attachment to the recipient of the message. Additionally, the EntShare object was used to render the URL preview to the sender of the message, when the sender subsequently viewed the sent message in his or her inbox. In the [redacted] (FB000027063-27067) in the Facebook code, the function [redacted] at lines 28-31 was called to render the attachment, and that function took an EntShare object as a parameter. That function in turn, called t[redacted] (lines 12-19) to perform a data conversion, and then called [redacted] (lines 43-238) to perform the actual rendering. For a given thread in a user's inbox, the code in [redacted] (FB000027056-27062) was used to render all the messages in the thread. The function [redacted] defined at lines 46-257, then iterates over the messages to be rendered and, at lines 101-102, calls r[redacted] (mentioned above) to render the share attachment. If an EntShare object was not created on the server when the message was sent initially (which was often the case, as discussed elsewhere in this report), the URL preview could not have been displayed to the recipient or to the sender. (*See also* January 14 Himel Decl. ¶¶ 16-18.)

42. When an EntShare object was created, the code would have attempted to store within the EntShare object, among other things, the numeric Facebook ID of the person who sent the message and which URL was specified by the URL attachment (regardless of whether the sender decided to keep the URL in her message or not). These pieces of data are required in order to display the attachment to the sender or recipient when they view their messages, and storing them together in an object allows for more efficient code and more uniform code operation. Additionally, in the case of messaging, after the EntShare object was created, the Facebook server would likely have executed code to increment the "share_count" field in the

EntGlobalShare object corresponding to the URL specified by the URL attachment. No person-specific information about the message received by the server was incorporated in the EntGlobalShare object, including in the "share_count" field (which is just an anonymous, aggregate counter). In contrast, if a sender posted a URL attachment in a post (which may have been visible to the general public), the Facebook ID of that person may have appeared in the EntGlobalShare as an "association." However, incrementing the counters was a separate process, and there is no way to identify the specific list of actions (or people whose actions were) included in the counters. (*Id*. ¶¶ 25-28.)

B. [redacted] and Nectar Logging

43. Like other complex websites, Facebook logs and records actions taken and information shared on its site. As noted above, Facebook must store the information shared on its site, at the very least in order to display that information when people use features that require it (e.g., viewing your messages inbox or a social plugin counter). Complex websites also log activity on their sites (with varying frequency and degree of detail) for other purposes, including maintaining site integrity and operability and detecting and deterring abuse. I understand that two of Facebook's logging practices have been challenged in this case: logging share scrape actions through its Nectar platform and logging share actions in a table called "[redacted]

44. The [redacted] table, in Facebook's Hive database, contained data regarding share objects that was, at one time, used for an algorithm that informed the Facebook Recommendations social plugin. However, I understand from Facebook engineers that in 2011 (prior to the start of the Class Period), the table and its data were deleted altogether. (*See* January 14 Himel Decl. ¶¶ 44-48).

45. Nectar is a library, *i.e.,* a collection of scripts, that moves data from one Facebook system to another; this can include sending statistics to certain databases. I understand that it is

one of several tools used to log activity or data generated in the operation of Facebook in order to, for example, (1) ensure site integrity and security, (2) detect and prevent abuse, (3) research and test new and old products, features, and functionality, (4) detect and fix bugs, (4) balance traffic and load processing, and (5) learn about aggregate preferences and patterns to inform site improvements and feature development (among other things). (*Id*. ¶ 51.) When the unit of code that performs a share scrape (discussed above as part of the generation of a URL preview) is executed, the code may attempt to log the event using Nectar. More specifically, when a URL preview is generated for a Facebook message prior to sending the message, if the components for the preview are not already on a Facebook server, and instead the server sends a request to scrape the third-party site, the code may have attempted to log the event using Nectar. However, my review of the code indicates that Nectar logged instances where the share scraper code was executed only on a sampled basis. (*See* [redacted] (FB000027137-27148) at lines 101-139; [redacted] (FB000027118) at lines 29-38; [redacted] (FB000027111-27117) at lines 45-111, 146-161, and 254-267; [redacted] (FB000027119-27129) at lines 246-280, 307-327,348-355.) Specifically, within [redacted] (FB000027137-27148), the sampling rate was retrieved by the [redacted] (lines 101-139), such that the sampling rate was either a single number or a mapping of URL domains to sampling rates. In other words, the sampling rate could be customized for specific URL domains. (*See* [redacted] (FB000027137-27148) at lines 120-131.) Ultimately, this sampling rate was used by the [redacted] at lines 307-327 of [redacted] (FB000027119-27129) to determine if a given scrape is to be logged. (*See also* FB000014195; January 14 Himel Decl. ¶¶ 52-55.) My review of the code indicates that it was configured to allow the sampling rate to be changed at any time, and I understand

from the Facebook engineers that it was in fact changed over time. Between June 13, 2011 and June 20, 2012, the Nectar sampling rate was generally set at 1%, and following June 20, 2012, it was generally set at 0.01%. (*See* January 14 Himel Decl. ¶ 52.) Further, I understand from Facebook engineers that URL shares for certain sites were sampled at a different rate at certain points in time in order to aid with debugging issues. (*Id*.) Accordingly, whether a user's inclusion of a URL in a draft message actually resulted in a share scrape that was logged in Nectar depended not only on whether there was a successful scrape, *but on the specific sampling rate associated with that URL at that point in time.*

46. Also, both the execution of the share scrape function and the logging of the share scrape event in Nectar occurred before the sender sent the draft message. (*See* [redacted] (FB000027137-27148); *see also* January 14 Himel Decl. ¶ 54.)

VIII. RESPONSE TO THE GOLBECK REPORT

47. I have been asked by counsel to respond to certain portions of the Golbeck report as augmented by Dr. Golbeck's deposition testimony. In this section, I provide my responses as well as the bases for my disagreements with Dr. Golbeck. I understand that the opinions I express in this report will be considered by the Court for class certification purposes; thus, it is not my intent to exhaustively respond to Dr. Golbeck with respect to topics that are unrelated to class certification. I reserve the right to respond to the Golbeck Report on such topics in the future if asked to do so.

A. The Alleged "Interceptions"

48. Dr. Golbeck refers to several routine computer programming functions as "interceptions." I offer no opinion at this time on whether the practices Dr. Golbeck discusses could be deemed "interceptions" for purposes of wiretapping law (though it would seem odd to

me if they could be). As discussed below, however, there was considerable variation in these practices and in my opinion it would be necessary to conduct a message-by-message analysis to determine whether the purported "interceptions" occurred for any given message.

1. Creation of Share Objects and Incrementing Global Objects

49. Dr. Golbeck characterizes as an "interception" or "redirection" the creation of a data structure, namely the EntShare object discussed above, when a message with a URL attachment is received on and processed by a Facebook server. (Golbeck Report ¶¶ 42, 55.) As someone who has been working and teaching in the field of computer science for more than 30 years, I disagree with these conclusions and the use of this terminology. As discussed above, the creation of "objects" is a standard part of object-oriented programming, and there is nothing unusual about Facebook's creation of EntShare objects from URL attachments. Indeed, as noted above, many objects were often created in connection with the processing of a Facebook message (such as "EntMessages" and "EntMessageThreads") and Dr. Golbeck does not contend that the creation of these other objects was an "interception" or "redirection" of a message. If Dr. Golbeck's characterization of an "interception" or "redirection" based on the creation of data structures (objects) were correct, there would be no way for the provider of a service involving communication between people to know when it was or was not performing an "interception."[3]

50. At her deposition, Dr. Golbeck attempted to distinguish the EntShare objects from other objects created during the processing of a Facebook message on the basis that Facebook was "collecting information that's in that URL attachment" and "directing that to code that is not part of the message delivery process." (Golbeck Deposition Transcript ("Golbeck Dep.")

[3] In her report, Dr. Golbeck says, "In the case of Private Messages, the EntShare represents a URL detected by Facebook software monitoring the keystrokes typed by the user while composing a private message." (Golbeck Report ¶ 40.) This is incorrect. As noted above, share objects (represented in "EntShares") represent URL attachments to posts and messages (in certain circumstances).

249:25-250:21.) As discussed above, however, Dr. Golbeck is incorrect in stating that the code that creates the EntShare object is not part of the message delivery process. As seen in [redacted] (FB000027063-27067), the EntShare object for a URL attachment to a Facebook message is used in rendering (displaying) the corresponding URL preview to the recipient of the message. Additionally, the EntShare object is used to render the URL preview to the sender of the message when the sender subsequently views the sent message in his or her inbox. (*See* [redacted] (FB000027056-27062).)[4]

51. At her deposition, Dr. Golbeck also stated that she was "very surprised" that Facebook was "creating metadata" (which is data about data) on communications among people using Facebook. (Golbeck Dep. 243:11-254:9.) "I was very surprised at the fact that they were creating metadata in a way that's totally unnecessary for that message to get delivered when I sent a private message." (*Id.*) However, as described above, the Entshare object Dr. Golbeck is discussing *is* used in the delivery of the message to the recipient, specifically in the rendering of the URL preview to the recipient. Additionally, as a computer scientist, I do not find it surprising that Facebook—or any provider of electronic communications between people—collects metadata concerning the communications. There are many reasons that a service provider would collect metadata about communications, including:

- collecting statistics about the geographical locations of senders and recipients to improve performance and reduce failures, e.g. by adding computing and

[4] Dr. Golbeck apparently maintains that it is not necessary *to design* a messaging system to rely on share objects to display URL attachments, but she testified that she is not aware of whether Facebook's actual implementation of its messaging system *does in fact* rely on share objects to deliver URL attachments and concedes that it may. (Golbeck Dep. 263:16-18 ("Is it the case that Facebook references that user share object? Potentially."); *id.* at 264:11-13 ("But I'm unaware of how Facebook might use that user share object for rendering on the recipient side.").)

communication resources to best reduce network congestion and balance the computing load on servers;

- saving web page images referred to in messages so that the web pages do not have to be repeatedly accessed at a cost of network congestion and server load; and
- for security by analyzing and saving patterns in the message content and traffic patterns to detect malevolent communications.

52. To the extent that Dr. Golbeck is characterizing the creation of an object (as opposed to other types of data structures) as an "interception" or "redirection," the creation of objects is a fundamental operation in all programs written in commonly used object-oriented languages, such as Java, C++, PHP, and JavaScript—the latter two being the languages used to implement the accused Facebook functionality.

53. Indeed, during her deposition, Dr. Golbeck could not identify whether URLs shared in posts (as opposed to messages) would be "interceptions." Despite admitting that the technology for creating share objects in the context of URLs in messages is the same across the Facebook platform, and "basically the same" as sharing URLs in public posts, she was not sure if shares in posts would be an interception because "[i]t's just such a different context." (Golbeck Dep. 256-257.)

54. Dr. Golbeck also states that she considers the "increase in the share_count in the EntGlobalShare's tracking_info field [to] constitute the interception, analysis, and use of the contents of user's Private Messages." (Golbeck Report ¶ 42.) I also disagree with this opinion. Because the "share_count" is a statistic that is kept about EntShares in messages—one of numerous statistics—with no person-specific information, and that is kept by the portion of the Facebook system that is actually supporting the messaging, it would strike one of skill in the

field of computer science as odd to characterize the maintaining of the "share_count" as an "interception" or "redirection" of any particular message data.

2. [redacted] and Nectar Logging

55. Dr. Golbeck also opines that Facebook's "logging" of some URLs shared in messages constitutes an "interception." (*Id.* ¶¶ 43-54.) This is another very unusual opinion because logging is a process performed by virtually all software systems and occurs for many reasons, including the recording of errors, interactions, resource usage, security issues, etc. If simply logging various aspects of person-generated data as part of providing a service to people can be considered an "interception," service providers would not be able to determine which logging might be considered an interception and which might not be.

56. Dr. Golbeck's report cites to a source code file called [redacted] and claims that URL shares in messages were and continue to be logged in the [redacted]" Hive table referenced therein through the present day. (*Id.* ¶¶ 51.) I understand from Facebook's engineers that the [redacted] table to which Golbeck refers was deleted in 2011 prior to the Class Period and no longer exists. (*See* January 14 Himel Decl. ¶¶ 44-48.) Accordingly, no URL shares from messages would have been logged or stored in this table after that date. (*Id.*)

57. Dr. Golbeck's report also points to the logging in Nectar of a URL scrape or access to a URL that has already been scraped as using message data that has been intercepted. (Golbeck Report ¶¶ 54-55.) However, URL scraping or access to a pre-scraped URL occurs *before* the sender sends the message, and thus cannot be an "interception" of a sent message. At her deposition, Dr. Golbeck stated that logging in Nectar occurs before the message is sent. (Golbeck Dep. 306:24-307:12.) Thus, logging in Nectar that is triggered by the URL scrape or

access to the pre-scraped URL is irrelevant to Dr. Golbeck's assertions concerning the interception of message data or the use of data that she asserts has been intercepted.

3. Share Scraper/Web Crawler

58. I understand that Plaintiffs have previously referred to Facebook's share scraper functionality—which (i) retrieves a web page using the URL typed by a sender into a message, (ii) displays a preview of the web page to the sender, (iii) adds a URL attachment to the message, and (iv) creates a global share object representing the URL—as a "web crawler" (Golbeck Report at n. 10). As discussed above and as Dr. Golbeck admitted at her deposition (Golbeck Dep. 233:4-234:8), this entire URL scraping process occurs before the message is sent, and therefore it cannot even be accused of being an interception.

4. Code-Based "Devices"

59. Dr. Golbeck's Report also refers to portions of Facebook code as "code-based devices" or simply "devices." (Golbeck Report ¶¶ 17.b., 55). According to Dr. Golbeck, the above-referenced "interceptions" are accomplished through the use of these "code-based devices." In my 33 years as a computer scientist, I have never heard the term "device" used to refer to software. Within the field, "device" is used to refer to hardware—and I note that Dr. Golbeck does not identify any hardware device as performing the interception such that the device (such as a Facebook server) is not used in the ordinary course of Facebook's business. At her deposition, Dr. Golbeck admitted she had never used the term "code-based device." (Golbeck Dep. 206:21-207:2):

> Q. Have you ever used the term "code-based devices" in your academic career?
>
> A. I don't think so.

Q. Why not?

A. This is not the kind of thing I study. I don't write about code at all, I don't think, yeah.

As someone whose publications have all been about code and programming, I can confirm that "code-based device" is not a term of art used to refer to code alone.[5]

B. Variability in the Purported "Interceptions"

60. While I find Dr. Golbeck's characterizations of routine and unremarkable computer science practices as "interceptions" and "redirections" to be unusual, I do not offer any opinion at this time regarding whether these practices could be deemed to constitute illegal wiretapping. However, even if Dr. Golbeck were correct in her characterization of routine programming practices, such as EntShare object creation and URL logging, as "interceptions," determining whether such alleged interceptions actually occurred would require a message-by-message analysis, in the following sense:

- There were many circumstances where URL previews were not generated. (*See supra* paragraphs 33; January 14 Himel Decl. ¶ 20; Supp. Resp. to First Interrog. at 17-18; Adkins Decl. ¶¶ 8-9.) If a URL preview was not generated, neither a URL attachment nor an EntShare object could have been created. (January 14 Himel Decl. ¶ 16.)
- Even if a URL preview and URL attachment were generated, the sender had the option of deleting the URL attachment before sending the message. (*See supra* paragraph 29, 39; January 14 Himel Decl. ¶¶ 16, 18, 21.) Because no EntShare object would have been created if the sender deleted the URL attachment before sending the

[5] Additionally, I performed a web search for "code-based device" and, in all the references I found, the "device" is a hardware device that executed code.

message, there is no way to determine messages for which URL attachments were created and then deleted.

- The creation of a URL attachment and corresponding EntShare object for a Facebook message did not necessarily indicate that the message contained a reference to that URL. As described above, a URL attachment was generated and the corresponding URL preview was displayed while the sender was still typing and editing the message, prior to the message being sent. If the sender subsequently (but before sending the message) decided to remove the URL in the contents of the message, the URL attachment would *not* have been removed (unless the sender deleted it). In such an instance, the Facebook server code would have attempted to create the EntShare object for the URL after the message was sent, despite the fact that the URL did not appear in the text field of the message. In these cases, even under Dr. Golbeck's theory, the creation of the EntShare object due to a URL attachment could not be considered an interception of the Facebook message content, since the URL itself was not in the message content, and, in any event such a sender would not be within Plaintiff's proposed class as written (which requires a URL in the text of the person's message). Thus, a message-by-message analysis would be needed to determine which EntShare objects were created based on URL attachments for messages that actually contained the URL when sent versus those EntShare objects that were created based on URL attachments for messages that did *not* contain the URL. (*See supra* paragraph 30, 39; January 14 Himel Decl. ¶ 15.)
- The Facebook code—including the Sentry functionality—performs filtering of URL attachments to prevent malicious URLs from being transmitted to the message

recipient. In the cases where the Facebook code determines that a malicious URL is contained within the message or URL attachment, no corresponding Entshare object will be created. First, the code at lines 213-223 of [redacted] FB000027149-27162) invokes the Sentry code that will check if a URL is on a blacklist and will prevent the scrape from occurring. Second, the function [redacted] which is defined in [redacted] (FB000027068-27072) at lines 155-203 and is called in [redacted] (FB000027130-27134) at lines 156-167 to create the user share object, calls [redacted] at lines 179-202, setting the [redacted] to false. The function [redacted] is defined in [redacted] (FB000027073-27088) at lines 40-555, and, in the case the [redacted] is false, it invokes Sentry code at lines 94-142. (*See also* FB000014194; FB000014179; FB000014180-14182; Adkins Decl. ¶¶ 8-12, 16-23.)

- Even if an EntShare object was created based on a URL attachment of a sent message, the "share_count" of the EntGlobalShare object might or might not have been incremented (which Dr. Golbeck incorrectly considers to be an interception). As stated in Facebook's interrogatory responses (*see* Supp. Resp. to First Interrog. at 35; Sec. Supp. Resp. to Sec. Interrog. at 16) and the testimony of Facebook witnesses (*see* January 14 Himel Decl. ¶¶ 28, 40; September 25, 2015 Deposition of Ray He 77:10-78:23, 156:15-157:1), whether a "share_count" is incremented or not is affected by the intermittent hardware and software failures, race conditions, and data corruptions (e.g. incorrect mapping of URLs to EntGlobalShare objects) that occur in the Facebook system—to which any system of that size and scope is susceptible.

Because the software and hardware in a huge distributed system such as Facebook does not always operate reliably, I agree with Facebook that there is no way, in general, to determine whether a given message with a URL attachment resulted in the incrementing of the "share_count" in the corresponding EntGlobalShare object or not. Additionally, it is not possible, of course, to determine from the "share_count" if a given URL attachment contributed to the count.

- The same variability exists for Facebook's logging practices at issue. For people using Facebook who sent a message with a URL attachment, only those people for whom the URL attachment actually resulted in the successful creation of a share object (which, as described above, might never have been generated, for a variety of reasons) could potentially have had their URL share included in the "share_count" in the global share object or in a counter in the "link_stats" table or Graph API. If no share object was created (because, for instance, the URL attachment was deleted by the sender before sending the message, or the message or its attachments were blocked in the course of abuse- and security-related processing), neither the "share_count" nor any "link_stats" or Graph API count would have been incremented. And even when a share object was created, race conditions, database failures, and database contention each may have resulted in failure to increment the "share_count" and counters in "link_stats" and Graph API. (*See* January 14 Himel Decl. ¶¶ 27-29, 69, 73-78.) Further, global share objects were not created for Facebook webpages, and therefore if a Facebook webpage URL was shared in a message, that URL share could not have incremented the share_count in any global share object or in "link_stats" or Graph API. (*Id.* ¶)

- Logging to the [redacted] Hive table was subject to similar variability when it existed (prior to the Class Period). If there was no preview, or no share object, created from the URL in a message, it would not be logged to [redacted] And, various database errors and failures may have resulted in the URL share in that message not being included in [redacted]" (*Id.* ¶¶ 46-49.)
- This variability is even more inherent for Nectar logging of share scrapes. Because Nectar logging of share scrapes only occurred for a random sample of sharescraper events, and that the sampling rate itself varied by time and by URL (but remained between 0.01% and 1%), only a tiny fraction of people even may have been subjected to Nectar logging of share scrapes in connection with their messages that had a URL attachment, and there is no known way of tracing which share scrapes were logged in Nectar and which were not. (*See supra* ¶¶ 55, 57; January 14 Himel Decl. ¶¶ 52-57.)
- Moreover, as stated above, Dr. Golbeck has not examined any Facebook source code later than December 31, 2012, so she cannot tell from the Facebook code she examined how the "share_count" has been *used* by Facebook since that date. As Dr. Golbeck concedes (Golbeck Dep. 203:24-204:25), the features of Facebook that may have made use of anonymous, aggregate data partially based on URL attachments have evolved over time, and as discussed in greater detail below, there is considerable variability in the alleged uses. Furthermore, even during the period between 2009 and December 31, 2012, corresponding to the Facebook source code that Dr. Golbeck may have examined, there is no way to determine whether the internal counts were actually used in the various ways that Dr. Golbeck alleges, due to the various implementation issues discussed above (*e.g.* software and hardware failures, race

conditions, data corruptions) as well as the variability inherent in the basic functionality of those features (discussed further below). (*See* January 14 Himel Decl. ¶¶ 37-43, 60-65, 69-79; Declaration of Dan Fechete ("Fechete Decl.") ¶¶ 20-33, 43-44.)

61. In summary, because of the various scenarios under which an EntShare object may have been created for a message that might or might not have contained a URL and an EntShare object might or might not have been created for a message that did contain a URL, etc., it is not possible to use the creation of all EntShare objects from Facebook messages to determine generally which messages have been "intercepted" under Dr. Golbeck's theory of interception. That is, the determination of whether the creation of an EntShare object constituted an "interception" would need to be performed on a message-by-message basis. The same is true of Dr. Golbeck's assertions regarding "logging."

C. Variability in the Alleged "Uses" of "Intercepted" Data

62. For the reasons discussed in the Facebook fact declarations I have reviewed, the same variability that exists for the "interception" issue also exists for the "use" issue. For a variety of reasons, including changes of practices over time, it would be necessary to conduct a message-by-message inquiry to determine whether URLs shared in messages were used in the ways Dr. Golbeck states, and even then such a determination likely would not be possible. Further, I understand Dr. Golbeck acknowledged at her deposition that, other than one "edge case" that occurred inadvertently (and briefly) approximately one year before the proposed class period, the other "uses" of URL message shares Plaintiffs challenge involve the use of anonymous, aggregated data that cannot be used to identify, nor can be traced to, any specific person. (Golbeck Dep. 310:22-314:22.)

63. Examples of the variability inherent in any inquiry concerning each of the alleged "uses" include:

- **Insights, Related APIs, and Real-Time Analytics:** I understand that URLs sent in messages were included only in aggregated, anonymous counters visible to domain owners through Insights until October 11, 2012. Whether a given URL share in a messages was actually reflected in this aggregate demographic data during the proposed class period would have depended on the variability inherent in generating a preview and share object, as well as the inherent possibility of server and database errors, or load-time expiration, which could have led to a given URL share not incrementing the Insights counter at all. In addition, I understand that it would not be possible to determine whether any person ever actually used the API to view the aggregated share data for a given URL, since the owner of that domain might never have authenticated ownership and accessed Insights (an issue that cannot be retroactively ascertained that far back). I understand that the same is true for Real-Time Analytics – a specialized system designed to provide Insights data even more rapidly. It would not be possible to determine which (if any) proposed class members were impacted by this practice, given that it depended on the creation of share objects and their inclusion in the Insights counter. (*See* January 14 Himel Decl. ¶¶ 58-65.)
- **Plugin Counts:** The anonymously aggregated counts sometimes displayed with Facebook's social plugins could only potentially have been incremented based on URL shares in messages between August 16, 2010 and December 19, 2012. (*See id.* ¶¶ 37-43.) First, even during this time period, if a URL was shared in a message but no share object was created and/or the sharing of that URL did not increment the

"share_count" counter in the global share object, it would not increment any plugin count on the destination website. Second, if the destination website associated with the URL did not display a Facebook plugin count, or if the count was configured to display only that website's "fan" count (which was not configured to include URL shares from messages), the sharing of that URL would not have incremented the count on the website (even if a share object was created and the global share object's "share_count" was incremented). Third, the count on the destination website might still not have been incremented from a URL share in a message for other reasons (for example, if the URL included in a message was not exactly the same as the URL the developer passed to the plugin). Fourth, even if sharing a given URL in a given message did actually increment the count on the third-party website, there generally would be no way to trace which person or what particular action resulted in that increment. (*See id*.)

- **"Link_Stats" & Graph API**: "Link_stats" API queries and Graph API queries would have reflected the same anonymous, aggregate counters stored in Facebook's global share objects (or EntGlobalShares) but in a more aggregated form prior to October 16, 2012. After that, URL shares from messages would no longer have been included in the anonymous aggregate counts returned by a "link_stats" or Graph API query. Further, only those messages for which the URL attachment actually resulted in the successful creation of a share object could have had their URL share included in the results of a "link_stats" or Graph API query. Moreover, whether anyone ever actually accessed or called the "link_stats" or Graph API, and thus even saw this anonymous aggregate counter for any given URL that a sender may have shared in a

message during the Class Period, is an individualized inquiry and the answer will vary for different URLs sent by different purported class members. (*See id*. ¶¶ 73-79.)

- **Recommendations and Activity Feeds**: There likewise was considerable variability over time, during certain periods, and in a given instance of sharing a URL in a message, with respect to (i) which source code (either the "PHP back end" system or the "Taste" system) determined the URLs to be included in the Recommendations and Activity Plugin, (ii) whether and how that source code utilized information about URLs shared in messages, (iii) whether that source code could result in the display of a URL only shared in a message in the Recommendations or Activity Plugin, and (iv) whether a given URL had characteristics such that the source code would in fact display such a URL. For most of the proposed class period (from December 2011 to July 9, 2014) Recommendations and Activity Feeds, at most, considered URL shares from messages in their anonymous ranking algorithm in the rare instance that Taste failed and the PHP backend acted as a backup. It would not be possible to retroactively determine whether and which URL shares from messages actually contributed to the ranking algorithm in these rare instances. Then, from July 9, 2014, until June 23, 2015, after the PHP back end was discontinued, the Recommendations and Activity Feeds no longer considered URLs sent in messages at all. Additionally, during the entire proposed class period, it was not possible for the Feeds to ever actually display a URL only shared in messages (regardless of which back end system was used), and both the Recommendations and Activity Feeds were discontinued altogether on June 23, 2015. (*See* Fechete Decl. ¶¶ 23-34, 43-45.)

D. Continuing Conduct

64. Dr. Golbeck and I had access to the same Facebook source code produced in this matter, namely source code from the period of September 2009 to December 2012. However, the Golbeck report opines on the *current* operation of the Facebook software—without ever having reviewed current Facebook source code. One of skill in computer science would know there is no way to reliably determine the behavior of software—particularly when it comes to internal operations that are not visible to the sender or recipient, as is the case with the accused functionality—without examining source code. Thus, I do not believe Dr. Golbeck has a factual basis for the conclusions she reaches in Section V of the Golbeck report, entitled " Facebook's Conduct Continues to the Present" (Golbeck Report ¶¶ 94-97).

65. I also note that, in opining on several of Facebook's practices, Dr. Golbeck relies not on source code but on e-mails about potential practices. (*See, e.g.*, *id.* ¶¶ 46-49, 52-53, 66-71, 76-78, 86-93.) This is not a scientifically reliable manner on which to base conclusions about the operation of source code.

E. Lack of Ascertainability of Class Members

66. In her report, Dr. Golbeck says that "to retrieve a list of class members, the Code process should be relatively straightforward." (*Id.* ¶ 103.) In the next two paragraphs of her report, she provides "example" code that she contends would return a list of "Facebook user IDs of everyone whose actions had created an EntShare from a private message," and, in her deposition, she said that such a list would identify the class members. (Golbeck Dep. 331:2-8.)

67. That is incorrect. This query would return a list of people that is both under- and over-inclusive of the proposed class. In her deposition, Dr. Golbeck conceded each of these flaws in her proposed query:

68. This query will be under-inclusive in that it will not reflect recipients of messages with URL attachments. (*Id*. at 331:10-333:18.)[6]

69. This query will be under-inclusive in that it will not identify senders and recipients whose messages with URL attachments were deleted. (*Id*. at 336:11-14.)

70. This query will be under-inclusive in that it will not identify senders and recipients whose accounts were deleted. (*Id*. at 335:25-336:10.)

71. This query will be under-inclusive in that it will not identify senders whose messages were blocked for site integrity purposes. (*Id*. at 336:15-21.)

72. This query will be under-inclusive in that it will not identify senders whose URL attachment did not result in the creation of an EntShare object for any reason. (*Id*. at 337:1-16.)

73. This query will be under-inclusive in that it will not identify senders that deleted a URL attachment before it was sent. (*Id*. at 337:20-338:1.)

74. This query will be over-inclusive in that it will include senders whose messages did not contain URLs in their text. (*Id*. at 339:2-15.)

75. This query will be over-inclusive in that it will include senders who never typed a URL into a message, and instead merely chose to "Share" a URL through a "Share" button on a third-party website. (*Id*. at 339:18-340:13; January 14 Himel Decl. ¶¶ 9, 17.)

76. This query will be over-inclusive in that it will include senders and recipients outside the United States. (Golbeck Dep. 340:22-341:1.)

77. This query will be over-inclusive in that it will include senders of messages outside the Class Period. (*Id*. at 341:2-7.)

---

[6] In her deposition, Dr. Golbeck proposed that this flaw could be remedied by looking at each message to ascertain who the recipient was, a woefully infeasible methodology if done manually, or trying to draft code to do the same, but does not propose any actual methodology for doing so. She assumes this is feasible. (Golbeck Dep. 334:3-335:18.)

78. Moreover, Dr. Golbeck's query is overbroad in that it will identify senders that were not subject to the challenged "uses." In her deposition, Dr. Golbeck conceded each of these flaws in her proposed query and said that identifying those that were subject to the challenged "uses" would be "case-specific:"

79. This query cannot identify senders subject to Nectar logging. (*Id*. at 341:8-342:16.)

80. This query cannot identify senders whose shares incremented the [redacted] table. (*Id*. at 342:17-343:14.)

81. This query cannot identify senders whose share data was utilized by Taste. (*Id*. at 343:16-18.)

82. This query cannot identify senders whose share data was displayed in any Recommendations plugin. (*Id*. at 343:19-22.)

83. This query cannot identify senders whose share data was displayed in any Activity Feed. (*Id*. at 343:23-344:1.)

84. This query cannot identify senders whose share data was displayed in any API made available to third parties. (*Id*. at 344:2-6.)

85. This query cannot identify senders whose share data was displayed in Insights data made available to third parties. (*Id*. at 344:11-16.)

86. This query cannot identify people whose share data resulted in an increment in a social plugin count on a third-party website. (*Id*. at 346:1-6.)

87. Finally, I understand from Facebook engineers that this query would not correctly capture older objects in which certain necessary fields remained unpopulated. (*See* January 14 Himel Decl. ¶ 7.) Further, I understand that writing and executing even a corrected query would

be extremely burdensome and resource-intensive, if it were even possible, given the nature and breadth of Facebook's databases. (*See* January 14 Himel Decl. ¶ 8.)

F. Variability in Operation of the Source Code Across Senders and Recipients

88. As discussed in detail above, the Facebook code did not operate the same for all people who used Facebook messaging. Specifically, there is substantial variability due to: (1) the evolution of the software over time, (2) implementation issues (hardware & software failures, race conditions, data corruption, etc.) inherent in any large distributed system, (3) differences in the way that people used the messaging software (typing and then deleting URLs, deleting URL attachments, clicking on "Share" buttons, etc.), and (4) the behavior of third parties (e.g., in configuring their sites to display certain plugins or in accessing external APIs, as well as in the operation of site integrity processes, which compared messages and their attachments to other people's messaging patterns to try to detect abuse, which might impact whether the challenged practices occurred). (*See* January 14 Himel Decl. ¶¶ 20-22, 27-30, 37-43, 46-50, 55-57, 62-65, 73-79; Fechete Decl. ¶¶ 20-38, 43-44; Adkins Decl. ¶¶ 8-12, 16-23.)

89. For all the reasons given above, it is my opinion that the Facebook messaging software did not treat the people identified by Dr. Golbeck as being a "class" substantially the same.

IX. COMPUTER STORAGE

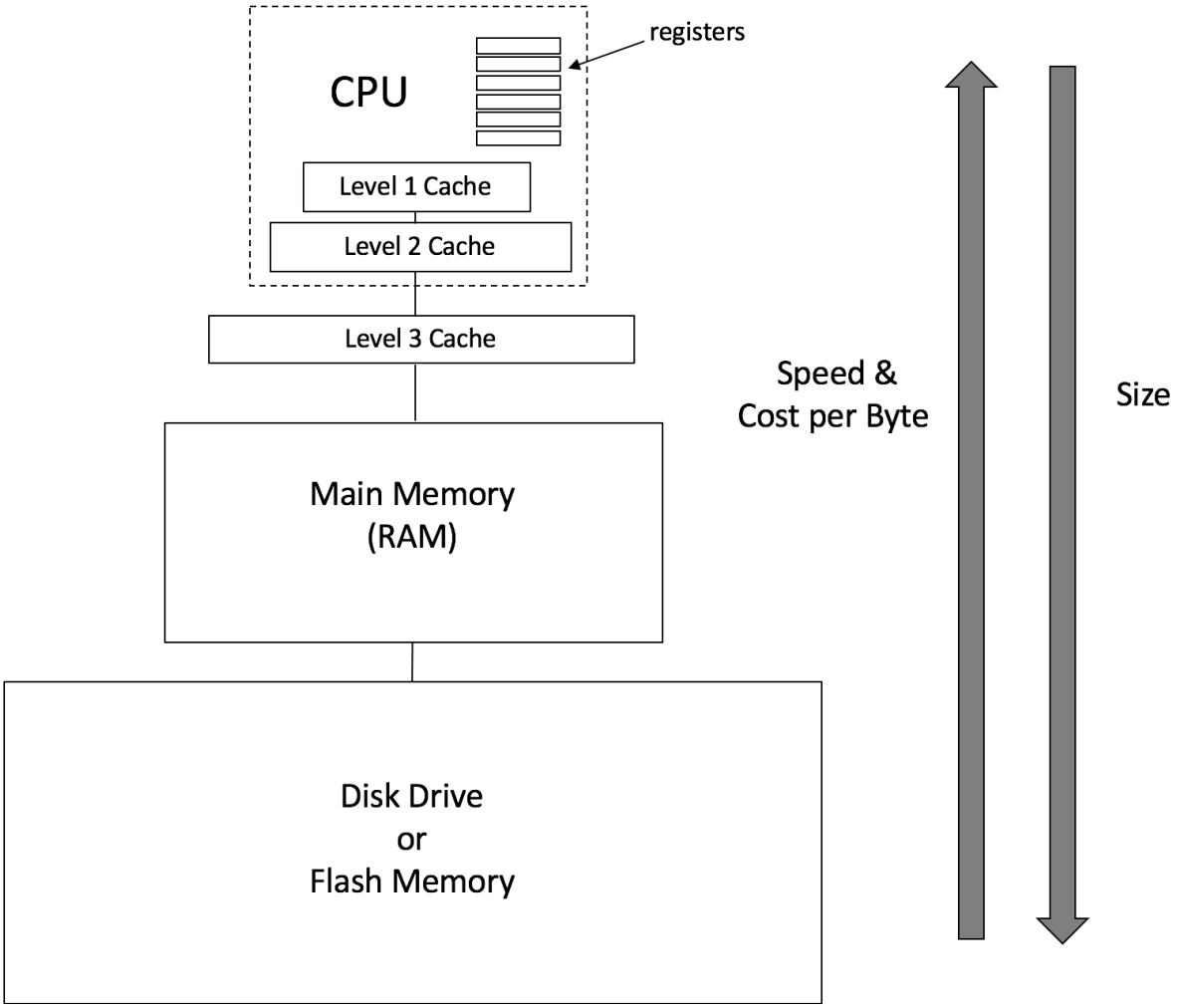
A. Computer Storage Technology

90. I have been asked to explain the meaning of "storage" in the context of computer science. Simply put, "storage" refers to the electronic components in a computer system that hold (store) data so that the data can be (1) processed by the central processing unit (CPU), *e.g.* used for arithmetic, image processing, text formatting, etc., (2) held temporarily until the data is

sent over a network or is sent to a display device, printer or other output device, or (3) kept for a longer period of time (perhaps indefinitely) so that it can be accessed in the future. The amount of data stored on a computer or other computing device (e.g. smartphone or tablet) is measured in terms of bytes, where a single byte comprises eight bits (each bit being a zero or a 1), which was the amount of data historically required to represent a single character of text (e.g. letter, digit, punctuation mark). On modern computing devices, storage capacity is normally multiples of megabytes (millions of bytes), gigabytes (billions of bytes), or terabytes (trillions of bytes). Naturally, mobile devices such as smartphones and tablets generally have less storage than personal computers, which, in turn, have less storage than large servers.

91. Computer systems have a number of different forms of electronic storage, ranging from small, fast, and expensive (on a per-byte basis) to large, slow, and inexpensive. Below is a diagram that I created to illustrate the so-called *memory hierarchy*, i.e. the different types of storage found on computer systems, including registers (very fast, very small storage within a CPU), cache memories (small, fast memories, often within the CPU), RAM (random access memory, aka "main memory"), and disk drive or flash memory.[7]

[7] Historically, there has been another level at the bottom of the hierarchy, archival storage – generally consisting of backup tapes or optical media (e.g. DVD-ROMs) – that is not shown.



As shown in the diagram, the storage types near the top of the diagram are small and fast, and the storage types near the bottom are larger and slower.

92. Storage can also be divided into two categories, "volatile" and "non-volatile." The term "volatile" means that the storage only retains the data while power is applied, so if the computer is turned off, the data is lost. Registers, cache memories, and RAM are examples of volatile electronic storage. The term "non-volatile" means that data is preserved in the storage even when no power is applied; thus, non-volatile electronic storage is often used for storing data for longer periods of time than volatile storage. Examples of non-volatile storage include disk storage, flash memory, magnetic tapes and optical media (CD-ROMs, etc.).

B. The Incorrect Distinction Between "Memory" and "Storage"

93. Dr. Golbeck appears to distinguish "memory" from "storage." With respect to the processing of a Facebook message, she states,

> Until that point, the message, and any URL attachment, are only held in memory, not in storage. Any information in a computer system is in memory when it is not in storage; otherwise, it could not be in the computer at all.

(Golbeck Report ¶117).

94. Dr. Golbeck also testified about the distinction between "memory" and "storage:"

> Q. What's the difference between memory and temporary storage?
>
> A. Storage is -- so memory allows you to operate on some data if it's in memory. Storage, temporary or permanent, is a place where information is stored that it can't be operated on.

(Golbeck Dep. 287:3-9.)

95. Dr. Golbeck's characterization of memory and storage—namely that memory is not storage—is incorrect for the following reasons:

- This is not an accepted characterization—in fact, I have never encountered this distinction between memory and storage before now. Dr. Golbeck seems to be referring to the fact that the instructions of a CPU (such as those to perform arithmetic or logical operations) can directly access some forms of storage, e.g., registers, cache, and RAM, but can only access other forms of storage, e.g., disk drives and flash memory, indirectly by first loading the data from disk or flash memory into RAM. Dr. Golbeck's distinction between memory and storage has nothing to do with whether data is *stored* or not, only how the data is *accessed*.
- As evidence that "memory" in the computing field is a form of "storage," the Microsoft Computer Dictionary defines memory as follows (emphasis added):

> **memory** n. A device where information can be stored and retrieved. In the most general sense, memory can refer to external *storage* such as disk drives or tape drives; in common usage, it refers only to a computer's main memory, the fast semiconductor *storage* (RAM) directly connected to the processor. See also core, EEPROM, EPROM, flash memory, PROM, RAM, ROM. Compare bubble memory, mass storage. (Microsoft Computer Dictionary, 5th Ed., 2002 p. 333)

As the dictionary states, "memory," used in its general sense, refers to all types of storage and, in its common narrower sense, "memory" is used to refer to RAM, which is fast semiconductor storage.[8] In either case, of course, memory is a form of storage.

96. For these reasons, and as discussed extensively above, while Facebook message data is being processed in RAM in order to facilitate the forwarding of the message and to create objects representing different aspects of the message, that message data is in *storage* as the term is used in computer science.

97. I understand that the Electronic Communications Privacy Act (18 U.S. Code § 2510) specifically defines "electronic storage" to include "any temporary, intermediate storage of a wire or electronic communication incidental to the electronic transmission thereof," and that communications in electronic storage are governed not by the Wiretap Act (18 U.S. Code § 2511) but by another law (the Stored Communications Act, 18 U.S. Code § 2701) not being asserted in this case. The Facebook server memory (RAM) used to store a Facebook message before it is forwarded is being used precisely as the "temporary, intermediate storage" defined by the Act to be "electronic storage."

98. As a factual matter, Dr. Golbeck is incorrect when she states that the message data is never in storage on Facebook servers until it reaches the recipient's mailbox – even given Dr.

[8] More recently the common use of "memory" is expanded to include "flash memory", which is a slower and cheaper form of semiconductor storage than RAM and has replaced disk drives in many computing devices.

Golbeck's definition of "storage" (which I disagree with). Upon reaching any Facebook server, a message is stored on the server's network card, which provides temporary storage until the message can be loaded into the server's RAM to be processed by the server. Since the server cannot operate on the message data while the data is stored on the network card, and the network card is clearly part of the server computer, the data would be in "storage" according to Dr. Golbeck's definition.

X. ORDINARY COURSE OF BUSINESS

99. Dr. Golbeck's report also discusses whether Facebook's alleged "interception" of URLs shared in messages was "necessary for Facebook to deliver private messages" (Golbeck Report ¶ 17.b.) and ultimately "conclude[s] that the interception, analysis, and use of URL shares in Private Messages is not necessary for the functionality of message sharing in Facebook." (*Id.* at ¶ 109.) However, my understanding is that whether or not the alleged interception of message content was "necessary" is not relevant to the "ordinary course of business" inquiry.

100. Rather, for purposes of assessing whether the "ordinary course of business" exemption to the Wiretap Act should apply, I understand that the Court has stated that it must determine whether the interception is "related or connected to [the] electronic communication provider's service, even if it does not actually facilitate the service…[and whether there is] some nexus between the need to engage in the alleged interception and the subscriber's *ultimate* business, that is, the ability to provide the underlying service or good." (Order on Mot. to Dismiss (Dkt. 43) 11-12.) Accordingly, Dr. Golbeck's test for assessing whether Facebook's alleged "interceptions" fall within the "ordinary course of business" exemption appears to be inconsistent with the Court's order, and I think her conclusions are irrelevant. Instead, Dr. Golbeck should have inquired whether there exists some nexus between the alleged

"interceptions" and Facebook's ultimate business, or its ability to provide its underlying service. Of course, Facebook's underlying service is *not* merely to provide a messaging product but to provide a massive social network comprising various features allowing people to share information and ideas and to engage and connect with other people in a safe, efficient online environment. Based on my review of the source code, I believe any alleged "interception" (if indeed, one exists, which I do not believe it does) more than qualifies under the "nexus" test set out by the Court, since it is related to both the messaging feature as well as other features that make up the Facebook service.

101. First, as discussed above, my review of the source code indicates that URL share objects (the creation and storage of which Plaintiffs claim constitutes an illegal interception) are used to display the URL preview to the recipient and sender viewing the message in their inboxes. This contradicts Dr. Golbeck's assumption that share object creation must not be necessary for sharing URLs in messages because Facebook concedes that share objects are not always created. (Golbeck Report ¶ 110.) Indeed, if no share object is created for a URL sent in a message (for any of the reasons explained above), the recipient will *not* be able to see the URL preview. Thus, in Facebook's source code—as written—the creation and storage of share objects is not only related to, but indeed necessary for, successfully rendering a URL attachment associated with a message. And, without a share object, there would be no increment in the internal share counters in the global share object and Insights system, and no logging in [redacted]

102. Second, share objects are used for various site-improving purposes, including monitoring site integrity and preventing abuse. (*See* Adkins Decl. ¶¶ 12-13; Deposition of Michael Adkins 75:5-76:1 ([redacted], Facebook's anti-abuse system, takes into account several

things, including [redacted]); *id.* at 118:13-119:2 (message statistics stored in share objects were used for [redacted]

103. Third, global share objects are used to create URL previews (across Facebook, for posts and messages alike) and also help provide efficiencies for programming and operability and reducing traffic across the Internet by storing all the components for generating URL previews for a given URL on a Facebook server, eliminating the need to repeatedly send requests to the website at the same URL for preview information (which can potentially cause that site to crash). (*See* January 14 Himel Decl. ¶ 11.)

104. Finally, Facebook's "logging" is specifically to facilitate its messaging and other services, by providing data points for internal research and site maintenance, data points for site integrity analysis, and, in the case of the internal counts in the "link_stats" table and Insights system, contributing to measures of engagement displayed with Facebook's social plugins and other features as part of its social platform extending to third-party websites, which help users identify information they may be interested in viewing and sharing.

XI. RESERVATION OF RIGHTS

105. I reserve the right to supplement or amend my opinions in response to opinions expressed by Plaintiffs' experts, or in light of any additional evidence, testimony, discovery or other information that may be provided to me after the date of this report. In addition, I reserve the right to consider and testify about issues that may be raised by Plaintiffs' fact witnesses and

experts at trial. I also reserve the right to modify or to supplement my opinions as a result of ongoing expert discovery or testimony at trial.

Dated: January 14, 2016

DR. BENJAMIN GOLDBERG