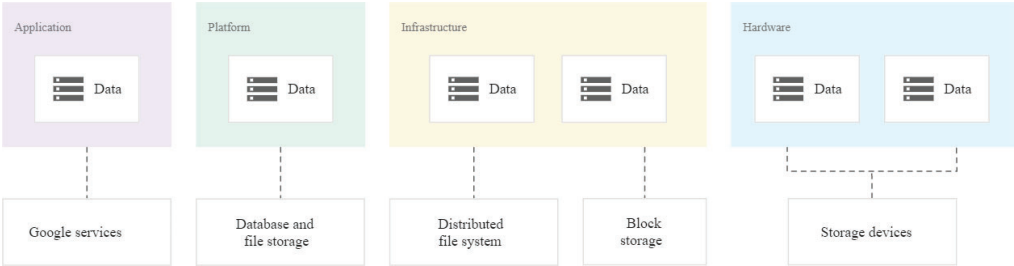
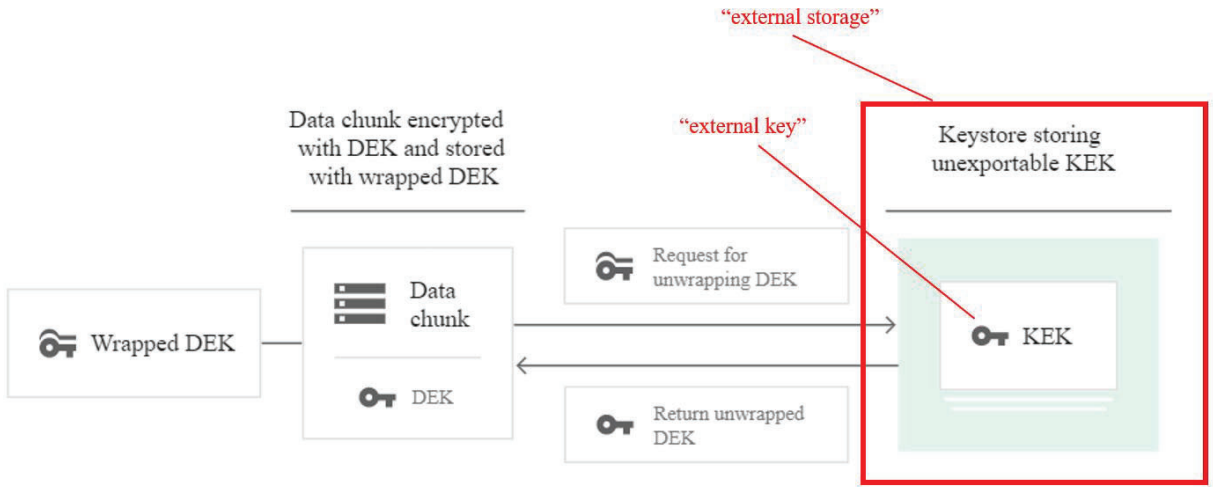
CHART A – U.S. PATENT NO. 10,452,854

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
Claim 1[pre] A method for securely storing a data set, the method comprising:[2]	Google's "Encryption at Rest" practices a method for securely storing a data set. For example, Google states its "comprehensive security strategy includes encryption at rest," and that it "encrypt[s] *all Google customer content at rest*, without any action required by" the customer. SFI-GOOGLE_000003146, *Default encryption at rest*, GOOGLE CLOUD (last visited Aug. 14, 2023) (emphasis added). "Google encrypts all customer content stored at rest, without any action from you, using one or more encryption mechanisms. The following sections describe the mechanisms that we use to encrypt customer content." *Id.*; *see also* SFI-GOOGLE_000005743, *Cloud Key Management Service deep dive*, GOOGLE CLOUD (last visited Aug. 14, 2023), https://cloud.google.com/docs/security/key-management-deep-dive (Cloud Key Management Service). Google securely stores data using "several layers of encryption that are generally used to protect user data in Google production data centers," SFI-GOOGLE_000003146 at -148, as seen in the image below.  *Id.*

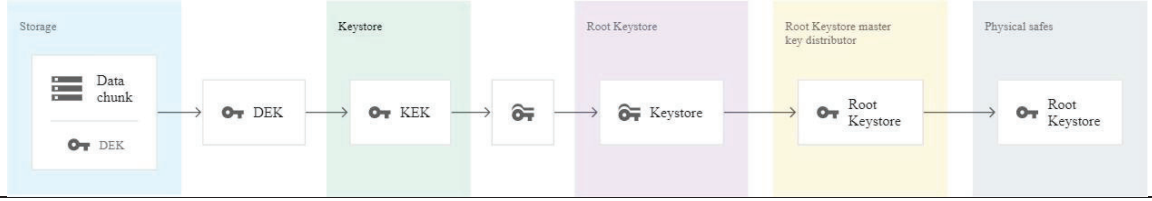
[1] As explained in more detail in the body of SFI's response to Interrogatory No. 7, each of the Accused Services stores customer data with the accused "Encryption at Rest" technology, and, on that basis, infringes.

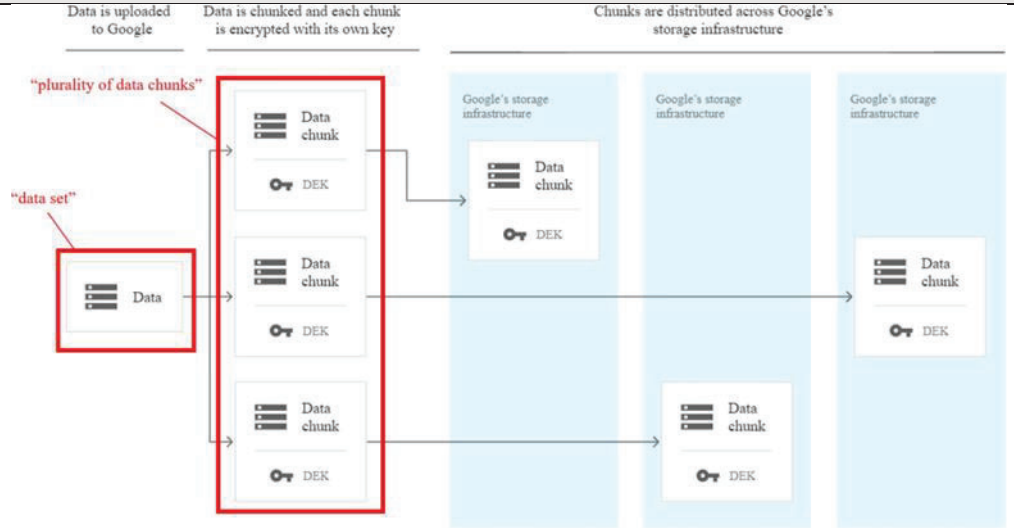
[2] Unless otherwise indicated, SFI contends that each element of every asserted claim of the '854 Patent is literally present in the "Encryption at Rest" utilized by each of the Accused Services. To the extent any element is not literally present, SFI contends such element is present under the doctrine of equivalents.

Google Exhibit 1011
Google v. SFI

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
1[a] receiving an external key from an external storage system,	Google's "Encryption at Rest" comprises receiving an external key from a storage system. By way of example, Google's encryption technology receives an external key from a storage system in the form of, for example, a Key Encryption Key ("KEK"), which is stored in an external storage system Google refers to as "Keystore." *See* SFI-GOOGLE_000003146 at -150 ("These KEKs are stored centrally in Keystore, a repository built specifically for storing keys."); *see also* SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an exemplary annotated image from the Google Cloud webpage showing an example of an external key from a storage system[3]:  SFI-GOOGLE_000003146 at -153. This external key is received by, for example, the Root Keystore. Google states that "Keystore is protected by a root key called the *keystore master key*, which wraps all of the KEKs in Keystore. This keystore master

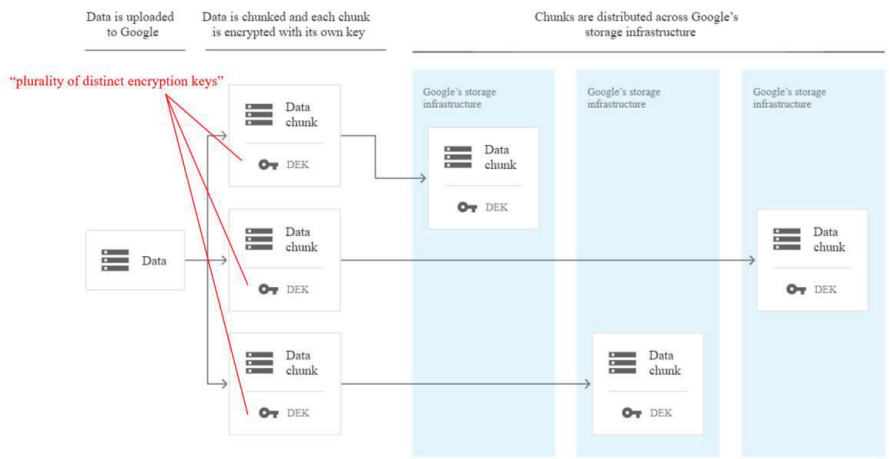
[3] The red boxes and accompanying text within images are annotations added by SFI for purposes of this claim chart.

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
	key is AES-256 and is itself stored in another key management service, called Root Keystore." *Id.* at -153. *See also id.* at -155 ("Keystore keys are wrapped with the Keystore master key, which is stored in Root Keystore."). The Root Keystore therefore receives the KEK in order to wrap and return it to Keystore. *See id.* at -154: 
1[b] generating a plurality of data chunks based on the data set, such that the data set can be reconstructed using at least a minimum number of the plurality of data chunks, wherein generating the data chunks comprises:	Google's "Encryption at Rest" comprises generating a plurality of data chunks based on a data set, such that the data set can be reconstructed using at least a minimum number of the plurality of chunks. For example, Google generates a plurality of data chunks based on a data set by splitting a user data set into data chunks. Google states that "[d]ata is broken into subfile chunks for storage." SFI-GOOGLE_000003146 at -148; *see also* SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an exemplary annotated image from the Google Cloud webpage showing an example of a data set and a plurality of data chunks:

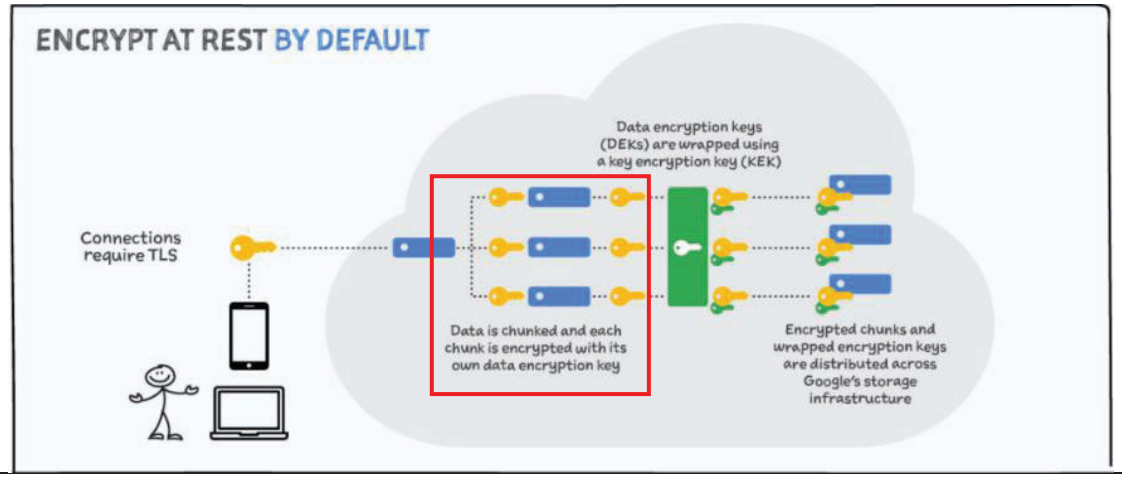
U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
	 SFI-GOOGLE_000003146 at -148. The data set can be reconstructed using at least a minimum number of the plurality of chunks. To do this, a Google application or service makes a call to the storage system. The storage system then "identifies the chunks in which that data is stored (the chunk IDs) and where they are stored," "pulls the wrapped DEK that is stored with [each] chunk," unwraps the DEK using the KEK, and Keystore "passes the unwrapped DEK back to the storage system, which decrypts the data chunk" and passes it to the requesting service or application. SFI-GOOGLE_000003146 at -152-153. Once "all the chunks that compose a specific file or data structure are decrypted, . . . the data [is] available to the requesting application." SFI-GOOGLE_000003287 at -293, *How Google Workspace uses encryption to protect your data* (Aug. 2020). Google has also explained the operation of this system as follows: > So first, how does [decryption and access of data] work in practice? So you can think about this as being three different actors here. There's a service that's using the data. There is a

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
	storage system that's storing the encrypted data. And there's a KMS that's storing . . . the key encryption keys. So what happens in practice is the service asks the storage system for some object. The storage system verifies that the service has the right to access that object, figures out all the chunks in which that data is stored, checks those ACLs. Then it pulls the data encryption keys that are sitting with those chunks of data, and passes those data encryption keys, encrypted, to the KMS. The KMS then verifies that the storage system--again, another ACL check--that it has the right to access those key encryption keys, decrypts those data encryption keys in memory, sends them back to the storage system. . . . Then from the storage system back to the service, the storage system decrypts the data, and sends back the plaintext data to the service, in most cases. In some cases, the service decrypts it directly. SFI-GOOGLE_000003191 at 8:47-9:59, Google Cloud Tech, *Managing encryption of data in the cloud (Google Next '17)*, YOUTUBE (Mar. 10, 2017), https://www.youtube.com/watch?v=StJ1NOQjAjo&t=365s.
1[c] distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set,	Google's "Encryption at Rest" comprises distributing the data set into a plurality of shares, wherein each of the shares comprises less than all of the data set. For example, Google distributes a data set into a plurality of shares when it "chunks" data for storage, and these shares comprise less than all of the data set. Google states that "[d]ata is broken into subfile chunks for storage." SFI-GOOGLE_000003146 at -148. A dataset uploaded by a Google user, for example, "will not be stored entirely in a single server, but instead will be broken up into chunks" and stored across different servers.[4] *See also* SFI-GOOGLE_000005743 (Cloud Key Management Service).
1[d] accessing a plurality of distinct encryption keys,	Google's "Encryption at Rest" comprises accessing a plurality of distinct encryption keys. For example, Google states that "[e]ach chunk is encrypted at the storage level with an individual data encryption key (DEK): two chunks won't have the same DEK, even if they are owned by the same customer

[4] SFI-GOOGLE_000005992, Google Cloud Tech, *How does encryption work at Google's data centers?*, YOUTUBE (Nov. 24, 2021), https://www.youtube.com/watch?v=82YD6esAK0o.

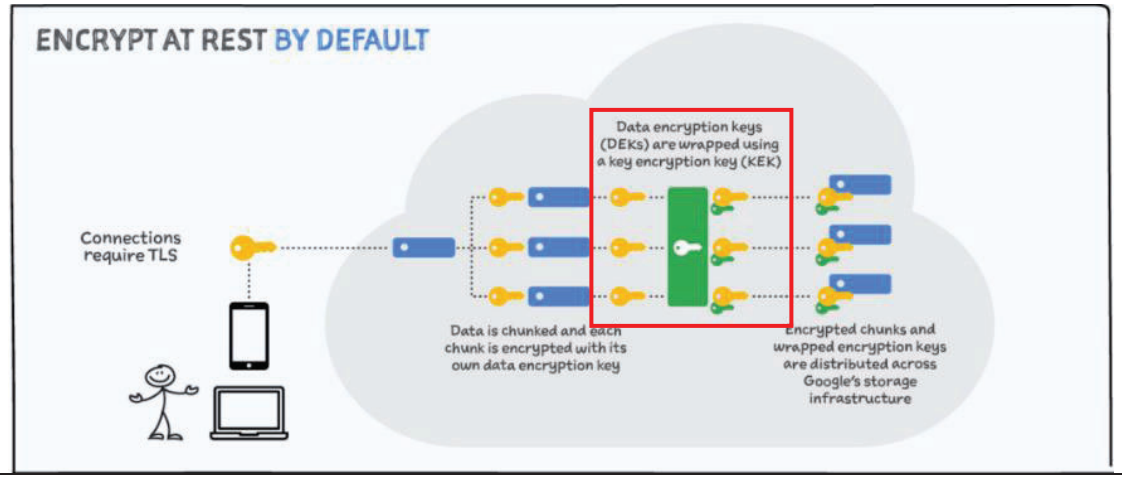
U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
	or stored on the same machine." SFI-GOOGLE_000003146 at -148; *see also* SFI-GOOGLE_000005743 (Cloud Key Management Service).[5] Google explains that "[t]he storage system generates DEKs using Google's common cryptographic library. In general, DEK[s] are then sent to Keystore to wrap with that storage system's KEK, and the wrapped DEKs are passed back to the storage system to be kept with the data chunks." *Id.* Below is an exemplary annotated image from the Google Cloud webpage showing an example of a plurality of distinct encryption keys:  As another example, during decryption, the DEKs are accessed when they are sent to Keystore to be unwrapped by the KEKs. SFI-GOOGLE_000003146 at -152-153; SFI-GOOGLE_000003287 at -293.

[5] *See also* SFI-GOOGLE_000005933 at 1:15-18 ("Each chunk is encrypted with a unique data encryption key.").

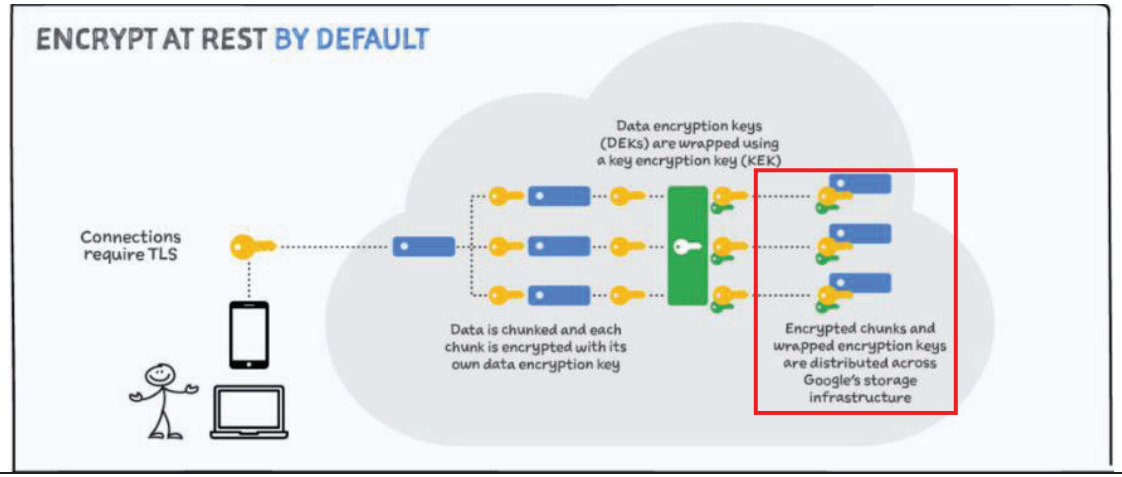
U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
1[e] encrypting each of the shares with a respective one of the plurality of distinct encryption keys,	Google's "Encryption at Rest" comprises encrypting each of the shares with a respective one of the plurality of distinct encryption keys. For example, Google encrypts each respective data share with a distinct encryption key when it "encrypts data as it is written to disk with a per-chunk encryption key." SFI-GOOGLE_000003287 at -293. Google states that "[e]ach chunk is encrypted at the storage level with an individual data encryption key (DEK): two chunks won't have the same DEK, even if they are owned by the same customer or stored on the same machine." SFI-GOOGLE_000003146 at -148-149; *see also* SFI-GOOGLE_000005743 (Cloud Key Management Service).[6] Below is an annotated image of the process of encrypting the data shares with distinct DEKs to generate data chunks in an encrypted state.[7] 

[6] *See also* SFI-GOOGLE_000005933; SFI-GOOGLE_000005992 at 1:15-18 ("Each chunk is encrypted with a unique data encryption key.").

[7] SFI-GOOGLE_000005816, *Data security in Google Cloud*, GOOGLE CLOUD: BLOG (July 20, 2022), https://cloud.google.com/blog/topics/developers-practitioners/data-security-google-cloud.

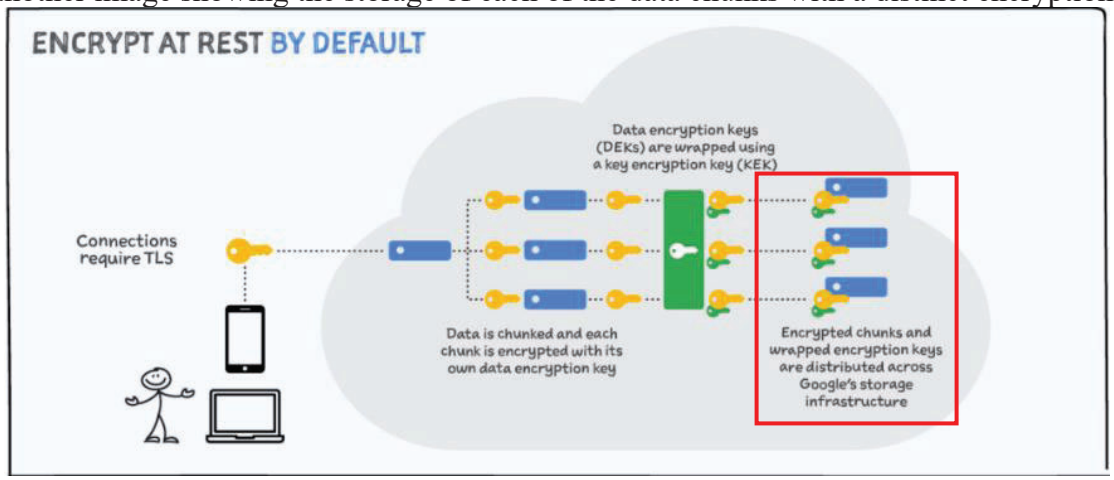
U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
1[f] performing an encryption operation based on the external key to further secure the plurality of data chunks; and	Google's "Encryption at Rest" comprises performing an encryption operation based on the external key to further secure the plurality of data chunks. For example, Google performs an encryption operation based on the external key when it encrypts each of the DEKs using a KEK. Google explains that "DEKs are encrypted with (wrapped by) a key encryption key (KEK)." SFI-GOOGLE_000003146 at -149. This further secures the data chunks, for in order for the data to be accessed, the KEK must first unwrap the DEK before the DEK can decrypt the data chunk. SFI-GOOGLE_000003146 at -153; SFI-GOOGLE_000003287 at -193; *see also* SFI-GOOGLE_000005743 (substantially the same for Cloud Key Management System). Below is an exemplary image showing the encryption operation of wrapping the DEKs using the KEKs.[8] 

[8] SFI-GOOGLE_000005816.

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
1[g] storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices.	Google's "Encryption at Rest" comprises storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys on a plurality of different storage devices. For example, after the original data is split and encrypted, each encrypted chunk is stored in a storage system with its data encryption key.[9] Google explains that "[e]ach chunk is distributed across our storage systems and is replicated in encrypted form for backup and disaster recovery." SFI-GOOGLE_000003146 at -149. Google explains that chunks of data are stored with their own DEK and are "stored across multiple servers."[10] *See also* SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an annotated image showing the distribution of data chunks and DEKs across Google's storage infrastructure.[11] 
Claim 2[pre] The method of claim 1,	Google's "Encryption at Rest" comprises the method of claim 1. *See* Claim 1, *supra*.

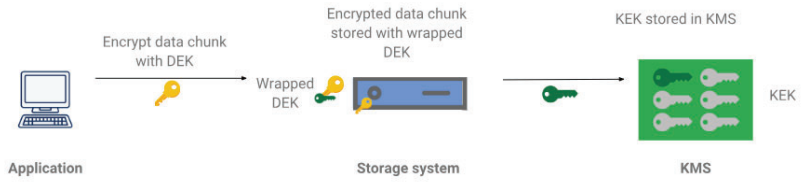
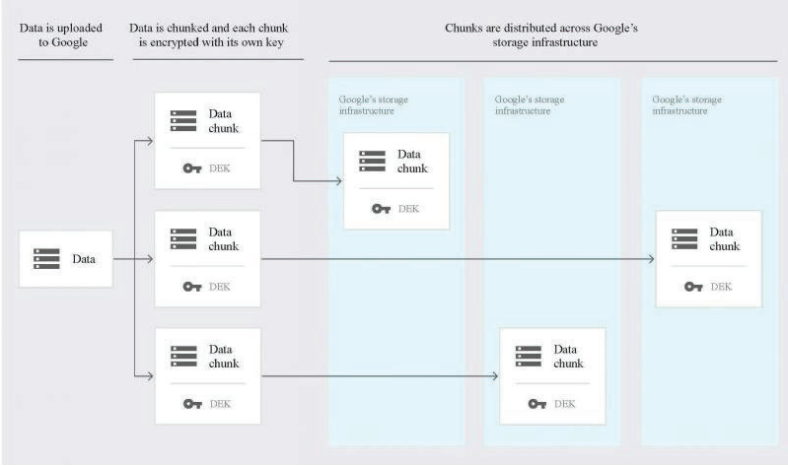
[9] SFI-GOOGLE_000005992.

[10] SFI-GOOGLE_000005992; *see also* SFI-GOOGLE_000005803, *Data and Security*, GOOGLE DATA CENTERS (last visited Aug. 14, 2023), https://www.google.com/about/datacenters/data-security/ ("Rather than storing each user's data on a single machine or set of machines, we distribute all data . . . across many computers in different locations.").

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
2[a] wherein storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys comprises storing each of the plurality of data chunks with at least one of the distinct encryption keys.	Within Google's "Encryption at Rest," storing with the plurality of data chunks data indicative of at least one of the distinct encryption keys comprises storing each of the plurality of data chunks with at least one of the distinct encryption keys. For example, Google stores with each data chunk of the plurality of data chunks data indicative of at least one of the distinct encryption keys by, for example, storing a DEK with each data chunk. SFI-GOOGLE_000003146 at -151 ("[W]rapped DEKs are passed back to the storage system to be kept with the data chunks."). Google explains that "[b]ecause of the high volume of keys at Google, and the need for low latency and high availability, DEKs are stored near the data that they encrypt." SFI-GOOGLE_000003146 at -150; *see also* SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is another image showing the storage of each of the data chunks with a distinct encryption key.[12] 

[11] SFI-GOOGLE_000005816.

[12] SFI-GOOGLE_000005816.

U.S. Patent No. 10,452,854	Google's "Encryption at Rest"[1]
	As another example, below is an image from Google Cloud's webpage illustrating that every encrypted chunk is stored with its wrapped DEK.[13]  Below is an exemplary image from the Google Cloud webpage showing DEKs stored with data chunks:  SFI-GOOGLE_000003146 at -149.

[13] SFI-GOOGLE_000005887, *Envelope encryption*, GOOGLE CLOUD (last visited Aug. 14, 2023), https://cloud.google.com/kms/docs/envelope-encryption.