

CHART B – U.S. PATENT NO. 11,068,609

U.S. Patent No. 11,068,609	Google’s “Encryption at Rest” ¹
Claim 1[pre] A method for securing data, the method comprising²	Google’s “Encryption at Rest” practices a method for securing data. For example, Google states its “comprehensive security strategy includes encryption at rest,” and that it “encrypt[s] <i>all Google customer content at rest</i>, without any action required by” the customer. SFI-GOOGLE_000003146, <i>Default encryption at rest</i>, GOOGLE CLOUD (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.” SFI-GOOGLE_000003146; <i>see also</i> SFI-GOOGLE_000005743, <i>Cloud Key Management Service deep dive</i>, 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. The diagram illustrates the layers of encryption at rest. It consists of four main colored boxes representing layers: Application (purple), Platform (green), Infrastructure (yellow), and Hardware (blue). Each layer contains one or more white boxes labeled 'Data' with a small icon. Below these layers are five white boxes representing underlying services: 'Google services' (under Application), 'Database and file storage' (under Platform), 'Distributed file system' (under Infrastructure), 'Block storage' (under Infrastructure), and 'Storage devices' (under Hardware). Dashed lines connect the layers to the services below them, indicating a multi-layered security approach. <i>Id.</i>

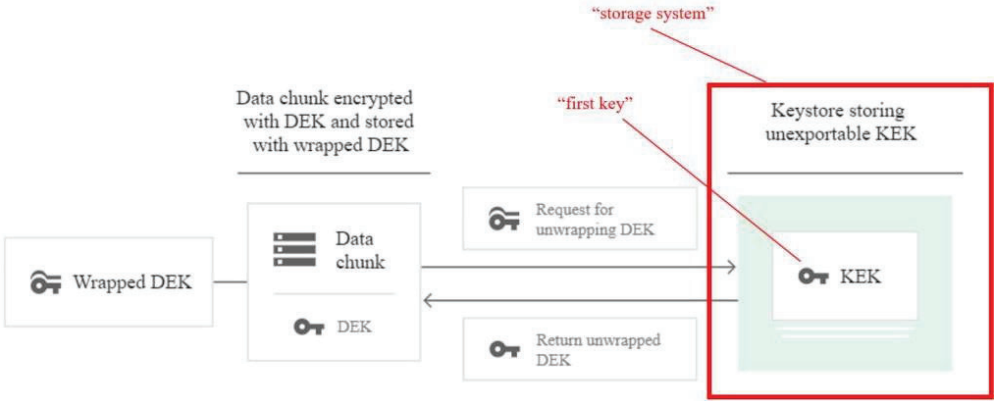
¹ As explained in more detail in the body of SFI’s response to Interrogatory No. 7, each of the Accused Services stores customer data with the accused “Encryption at Rest” technology, and, on that basis, infringes.

² Unless otherwise indicated, SFI contends that each element of every asserted claim of the ’609 Patent is literally present in the “Encryption at Rest” utilized by each of the Accused Services. To the extent any such element is not literally present, SFI contends such element is present under the doctrine of equivalents.

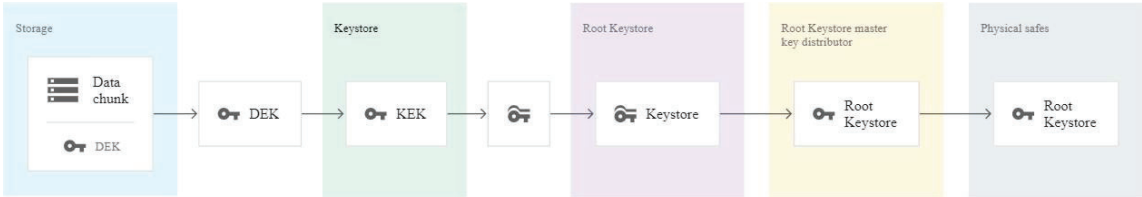
U.S. Patent No. 11,068,609	Google's "Encryption at Rest" ¹
1[a] executing code by a processor to perform:	Google's "Encryption at Rest" comprises executing code by at least one processor. For example, Google's default encryption at rest requires executing code by a processor such as the code associated with the wrapping and unwrapping of keys on Google's computer systems. Google has at least fourteen data centers in the U.S., each of which contains "many computers that store and process large amounts of information."³ Google's encryption at rest technology occurs on machines, executing code on processors, within Google's data centers.⁴ Below is an image depicting servers for, <i>e.g.</i>, storage and processing of encrypted data within a Google data center.  SFI-GOOGLE_000006005, Google Workspace, <i>Inside a Google data center</i> (Dec. 16, 2014), https://www.youtube.com/watch?v=XZmGGAbHqa0.

³ SFI-GOOGLE_000005806, *Google Data Centers*, GOOGLE (last visited Aug. 8, 2023), <https://www.google.com/about/datacenters>.

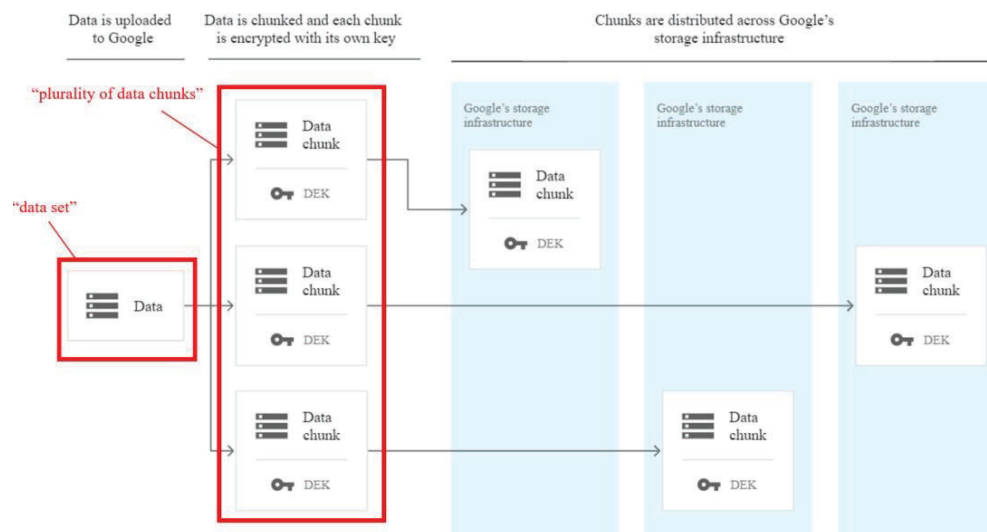
⁴ SFI-GOOGLE_000003146 at -155; SFI-GOOGLE_000005992, Google, *How does encryption work at Google's data centers?*, YOUTUBE (Nov. 24, 2021), <https://www.youtube.com/watch?v=82YD6esAK0o>.

U.S. Patent No. 11,068,609	Google's "Encryption at Rest" ¹
1[b] receiving a first key from a storage system;	Google's "Encryption at Rest" comprises receiving a first key from a storage system. By way of example, Google's encryption technology receives a first 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." <i>See</i> SFI-GOOGLE_000003146 at -150 ("These KEKs are stored centrally in Keystore, a repository built specifically for storing keys."); <i>see also</i> SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an exemplary annotated image from the Google Cloud webpage showing an example of a first key from a storage system⁵:  <i>See</i> SFI-GOOGLE_000003146 at -153. This first key is received by, for example, the Root Keystore. Google states that "Keystore is protected by a root key called the <i>keystore master key</i>, which wraps all of the KEKs in Keystore. This keystore master key is AES-256 and is itself stored in another key management service, called Root Keystore." <i>Id.</i> at -153; <i>see also id.</i> at -155 ("Keystore keys are wrapped with the Keystore master key, which is stored in Root Keystore.").

⁵ The red boxes and accompanying text within images are annotations added by SFI for purposes of this claim chart.

U.S. Patent No. 11,068,609	Google's "Encryption at Rest" ¹
	The Root Keystore therefore receives the KEK in order to wrap and return it to Keystore. <i>See id.</i> at -154.  <pre> graph LR subgraph Storage DC[Data chunk] DEK1[DEK] end subgraph Keystore KEK[KEK] end subgraph Root_Keystore [Root Keystore] KE2[Keystore] end subgraph Root_Keystore_master [Root Keystore master key distributor] RK[Root Keystore] end subgraph Physical_safes [Physical safes] RK2[Root Keystore] end DEK1 --> KEK KEK --> KE2 KE2 --> RK RK --> RK2 </pre>
1[c] generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein the data set can be reconstructed using at least a minimum number of the plurality of chunks;	Google's "Encryption at Rest" comprises generating a plurality of data chunks based on a data set, wherein each data chunk of the plurality of data chunks comprises less than an entirety of data of the data set, and wherein 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, wherein each data chunk of comprises less than an entirety of data of the 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; <i>see also</i> 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 plurality of data chunks:

Google's "Encryption at Rest"¹

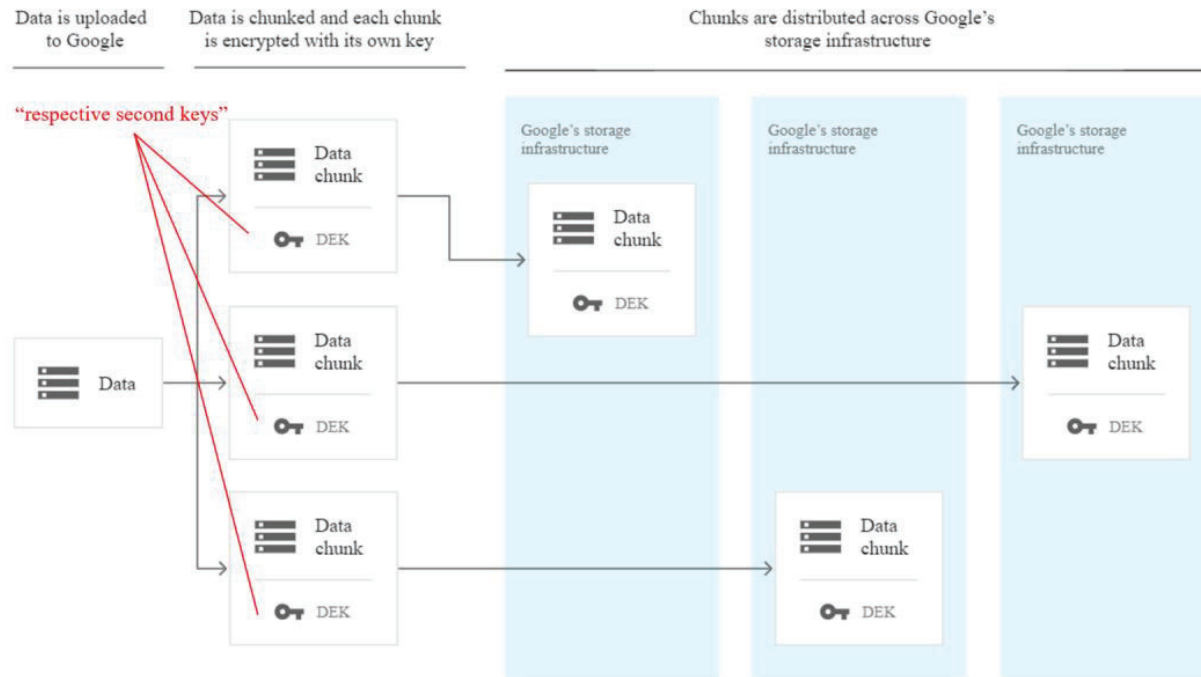


SFI-GOOGLE_000003146 at -148.

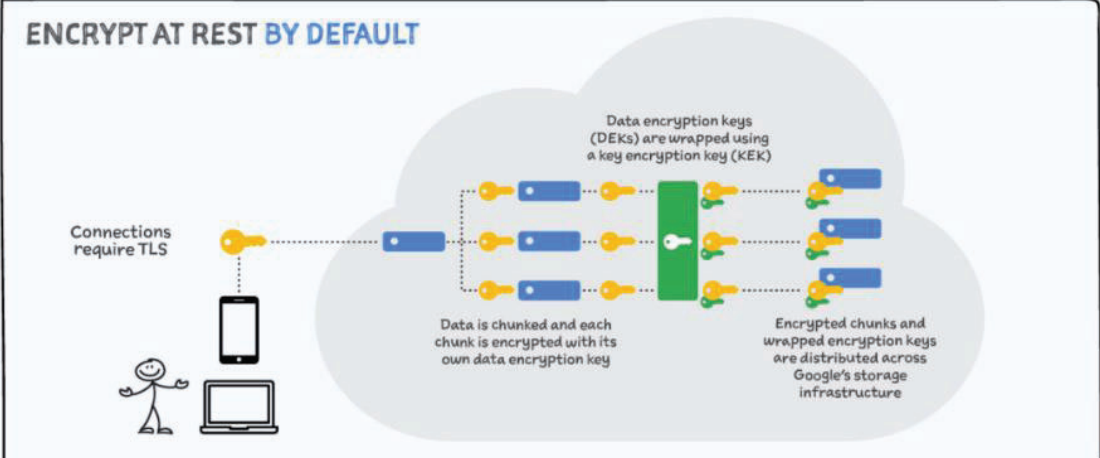
The data set can be reconstructed using at least a minimum number of the plurality of chunks. To do this, a Google application or service makes a call to the storage system. The storage system then "identifies the chunks in which that data is stored (the chunk IDs) and where they are stored," "pulls 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); see also SFI-GOOGLE_000003146 at -149.

U.S. Patent No. 11,068,609	Google's "Encryption at Rest" ¹
	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 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, <i>Managing encryption of data in the cloud (Google Next '17)</i>, YOUTUBE (Mar. 10, 2017), https://www.youtube.com/watch?v=StJ1NOQjAjo&t=365s.
1[d] encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other;	Google's "Encryption at Rest" comprises encrypting each respective data chunk of the plurality of data chunks with a respective second key, wherein each of the respective second keys are distinct from each other. For example, Google encrypts each respective data chunk with a distinct second encryption key. 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. 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." <i>Id.</i> at -151; <i>see also</i> SFI-GOOGLE_000005743 (Cloud Key Management Service).

Google's "Encryption at Rest"¹

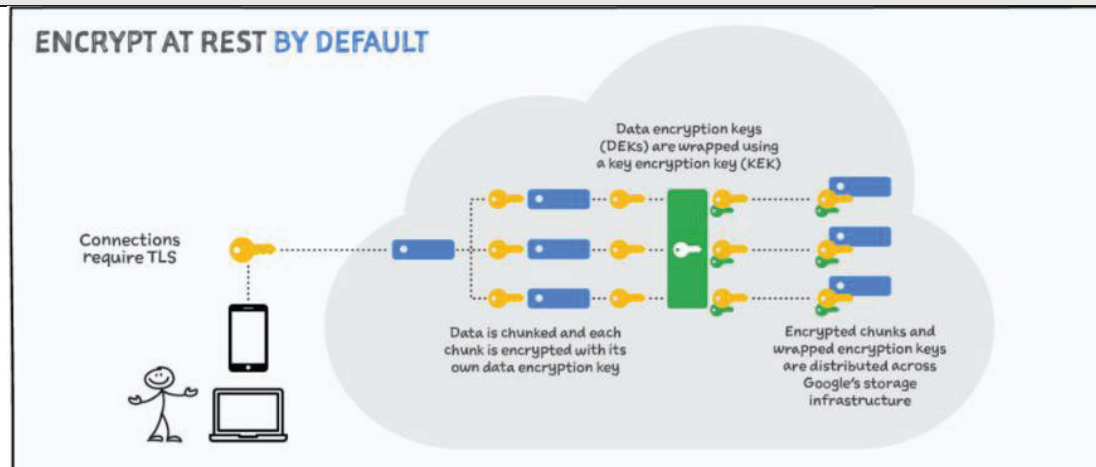


SFI-GOOGLE_000003146 at -149.

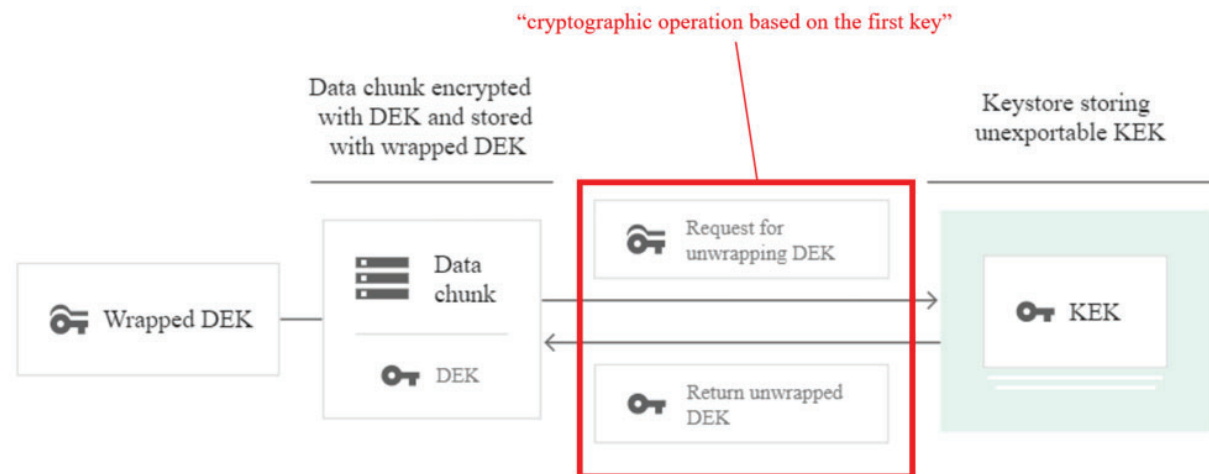
U.S. Patent No. 11,068,609	Google’s “Encryption at Rest”¹
	Below is another image of Google’s process of encrypting the data chunks with distinct DEKs.⁶ 
1[e] performing a cryptographic operation based on the first key to further secure the plurality of data chunks; and	Google’s “Encryption at Rest” comprises performing a cryptographic operation based on the first key to further secure the plurality of data chunks. For example, Google’s encryption at rest performs a cryptographic operation based on the first 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; <i>see also</i> SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an exemplary image showing Google’s cryptographic operation of wrapping the DEKs using the KEKs.⁷

⁶ 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>.

⁷ SFI-GOOGLE_000005816.



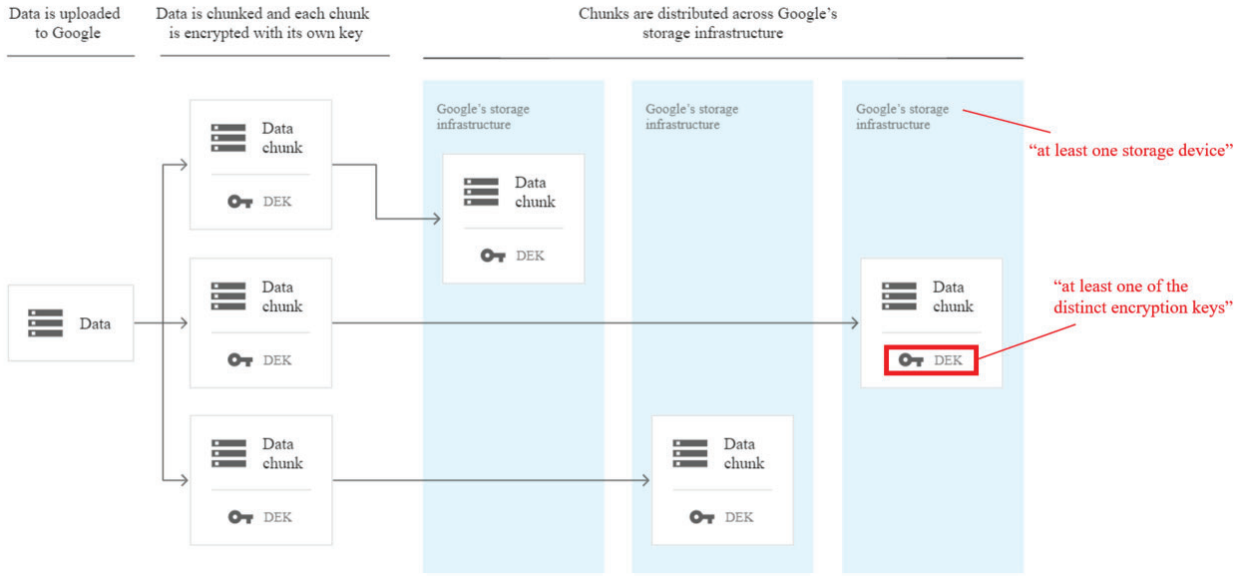
Below is exemplary annotated image from the Google Cloud webpage showing an example of a cryptographic operation based on the first key to unwrap the wrapped DEK:



SFI-GOOGLE_000003146 at -153.

U.S. Patent No. 11,068,609	Google's "Encryption at Rest" ¹
1[f] storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device.	Google's "Encryption at Rest" comprises storing, in a memory coupled to the processor, at least one data chunk of the plurality of data chunks with data indicative of at least one of the distinct encryption keys on at least one storage device. For example, Google stores, in a memory coupled to a processor within its data centers, the data chunks with data indicative of at least one of the distinct encryption keys on one or more storage devices, at least when it stores DEKs with the data chunks throughout Google's storage infrastructure. After the original data is split and encrypted, each encrypted chunk is stored in a storage system with its data encryption key.⁸ 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 states, for example, that in its "data centers, [encrypted] data belonging to Google Workspace customers is stored at rest in two types of systems: disks and backup media." SFI-GOOGLE_000003287 at -291; <i>see also</i> SFI-GOOGLE_000005743 (Cloud Key Management Service). Below is an exemplary annotated image from the Google Cloud webpage showing an example of the at least one storage device and data indicative of at least one of the distinct encryption keys:

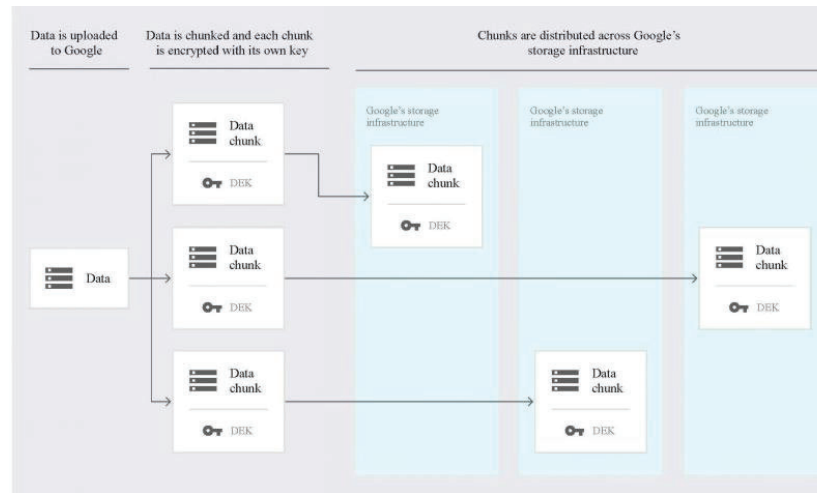
⁸ SFI-GOOGLE_000005992.

U.S. Patent No. 11,068,609	Google’s “Encryption at Rest”¹
	 The diagram illustrates the process of Google's "Encryption at Rest". It shows a flow from "Data is uploaded to Google" to "Data is chunked and each chunk is encrypted with its own key". The resulting "Data chunk" and "DEK" (Data Encryption Key) are then distributed across "Google's storage infrastructure". The diagram shows three storage infrastructure blocks. The first block contains a "Data chunk" and a "DEK". The second block contains a "Data chunk" and a "DEK". The third block contains a "Data chunk" and a "DEK". A red box highlights the "DEK" in the third block, with a red arrow pointing to it and the text "at least one of the distinct encryption keys". Another red arrow points to the "Data chunk" in the third block with the text "at least one storage device". SFI-GOOGLE_000003146 at -149.
Claim 2[pre] The method of claim 1,	Google’s “Encryption at Rest” comprises the method of claim 1. <i>See</i> Claim 1, <i>supra</i>.
2[a] further executing the code by the processor to perform: storing with each data chunk of the plurality of data chunks data indicative of at least one of the distinct encryption keys.	Google’s “Encryption at Rest” further comprises executing code by at least one processor to perform: storing with each data chunk of the plurality of data chunks data indicative of 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.”); <i>see also</i> SFI-GOOGLE_000005743 (Cloud Key Management Service).

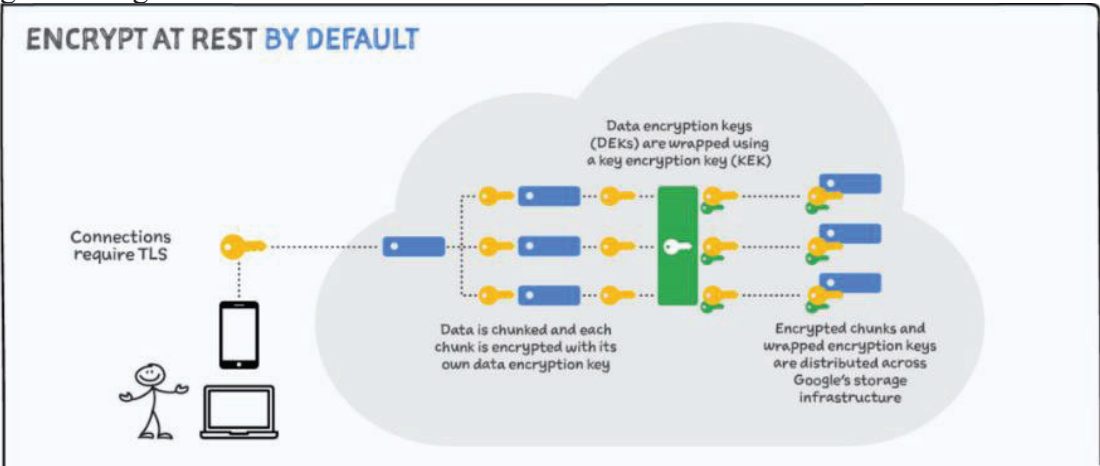
U.S. Patent No.
11,068,609

Google's "Encryption at Rest"¹

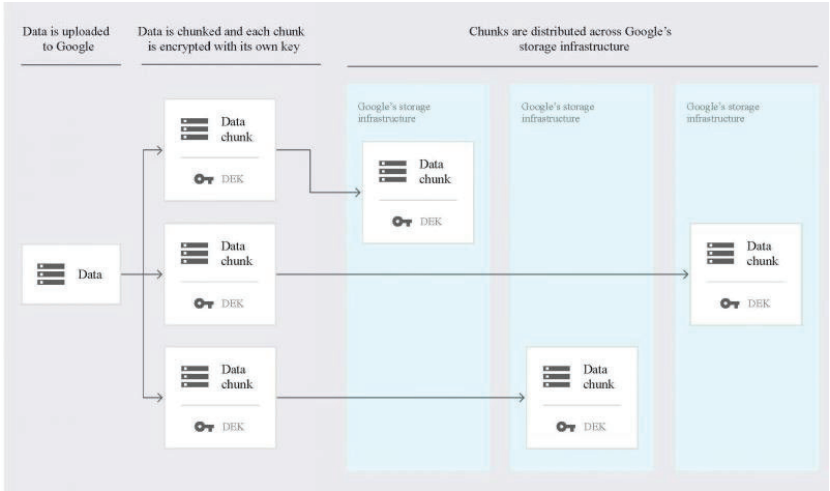
Below is an exemplary image from the Google Cloud webpage showing DEKs stored with data chunks:



SFI-GOOGLE_000003146 at -149.

U.S. Patent No. 11,068,609	Google's "Encryption at Rest" ¹
	Below is another image showing the distribution of encrypted chunks and wrapped encryption keys, stored across Google's storage infrastructure.⁹  The diagram, titled "ENCRYPT AT REST BY DEFAULT", illustrates the encryption process. On the left, a stick figure represents a user with a smartphone and a laptop. A dashed line labeled "Connections require TLS" connects the user to a cloud representing Google's storage infrastructure. Inside the cloud, a central green box represents the key management service. To the left of this box, several blue rectangles represent data chunks, each connected to a yellow circle representing a data encryption key (DEK). A label states: "Data is chunked and each chunk is encrypted with its own data encryption key". To the right of the green box, several blue rectangles represent encrypted chunks, each connected to a yellow circle representing a wrapped encryption key. A label states: "Data encryption keys (DEKs) are wrapped using a key encryption key (KEK)". A final label on the right states: "Encrypted chunks and wrapped encryption keys are distributed across Google's storage infrastructure".
Claim 3[pre] The method of claim 1,	Google's "Encryption at Rest" comprises the method of claim 1. <i>See</i> Claim 1, <i>supra</i> .
3[a] further executing the code by the processor to perform: storing with each data chunk of the plurality of data chunks data indicative of a distinct encryption key used to encrypt that data chunk.	Google's "Encryption at Rest" comprises further executing code by at least one processor to perform: storing with each data chunk of the plurality of data chunks data indicative of a distinct encryption key used to encrypt that data chunk. For example, each DEK stored with a data chunk is used to encrypt that particular data chunk. 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; <i>see also</i> SFI-GOOGLE_000005743 (Cloud Key Management Service).

⁹ SFI-GOOGLE_000005816.

U.S. Patent No. 11,068,609	Google’s “Encryption at Rest”¹
	Below is an exemplary image from the Google Cloud webpage showing how a DEK is stored with its corresponding data chunk:  The diagram illustrates the process of data encryption at rest in Google Cloud. It is divided into three main stages: 1. 'Data is uploaded to Google', where a 'Data' block is shown. 2. 'Data is chunked and each chunk is encrypted with its own key', where the data is split into three 'Data chunk' blocks, each associated with a 'DEK' (Data Encryption Key). 3. 'Chunks are distributed across Google's storage infrastructure', where the three encrypted data chunks and their corresponding DEKs are distributed across three separate 'Google's storage infrastructure' blocks. Each storage infrastructure block contains one 'Data chunk' and one 'DEK'. SFI-GOOGLE_000003146 at -149.
Claim 9[pre] The method of claim 1,	Google’s “Encryption at Rest” comprises the method of claim 1. <i>See</i> Claim 1, <i>supra</i>.
9[a] wherein the external key is an AES key.	Within Google’s “Encryption at Rest,” the external key is an AES key. For example, Google explains that “[f]or consistency, all KEKs are generated using Google’s common cryptographic library, using a random number generator (RNG) built by Google. This RNG is based on NIST 800-90Ar1 CTR-DRBG and generates an AES-256 KEK.” SFI-GOOGLE_000003146 at -151; <i>see also id.</i> at -152 (“DEKs are wrapped with KEKs using AES-256 or AES-128”); SFI-GOOGLE_000003287 at -293 (“The KMS wrapping keys that encrypt the chunk keys are 128-bit or stronger AES keys.”); SFI-GOOGLE_000005743 (Cloud Key Management Service).