The Simple Guide to Encryption Key Management

Understanding Common Data Privacy Methods and Misconceptions
The Simple Guide to Encryption Key Management

Understanding Common Data Privacy Methods and Misconceptions
Introduction

Congratulations. If you've made it this far, you're already on a safer path. You're steps away from a baseline knowledge of encryption and key management that will enable you to properly evaluate data privacy policies and technologies for yourself or for your business.

By reading this guide, here are some of the valuable takeaways that you'll gain – regardless of your background:

- You cannot determine the strength of an encryption technology without understanding how encryption keys are managed.
- There are 5 pillars of key management: key storage, policy management, authentication, authorization, and key transmission.
- Symmetric encryption can protect data at-rest, but asymmetric encryption is typically more effective when securely sharing data across different platforms.
- If vendors encrypt your data, they can usually still access it – along with hackers.
- For maximum security, you must exclusively host your encryption keys.

The following pages will explain them – along with several other topics – in greater detail, arming you with the key management expertise needed to stay safe in today's digital age.

*If you already have a firm grasp of cryptography basics, skip ahead to page 24 to learn about Virtru's unique approach to encryption key management.*
I. When Dreams Become Nightmares
Stop what you’re doing, and take a moment to dream. You just bought that white Porsche 911 you always wanted.

The top is down as you pull into your favorite restaurant. You hand the valet attendant your keys and head off to your meal.

You hear a loud screech once you walk in the front door. You run back outside, only to see the valet guy ride off with your car. The Flat 6 engine echoes as you stand there helplessly, wondering why you ever trusted someone else with something so valuable.
Okay, back to reality. You don't drive a Porsche, but you do share lots of emails, files, and other digital data. For many organizations, this data contains health, financial, and other sensitive information that's worth more than any car on the market.

Just as valet attendants park your car for you, technology vendors – such as Microsoft, Google, Amazon, and countless others – store your data. If you choose to encrypt your data, a vendor might also control the encryption keys protecting this info, just as a valet would for your car.

And like a locked Porsche 911, even encrypted data can be stolen if the keys get into the wrong hands.

This misconception has contributed to today's rise in cyber attacks. Individuals and companies know their online info is at risk and assume that encryption will keep it safe, but they don't realize how key management nuances can alter this assumption.

Where are encryption keys stored and transmitted? Who can access them? Where is encrypted content stored relative to these keys? How are recipients authenticated to decrypt the data shared with them?

Choosing an encryption method without answering these questions is like asking a stranger to park your car for you: things might end up okay, but without control over your keys and trust in where they're going, you have no way of ensuring that.
II. Why Key Management is Key
From a high level, the concept of encryption is simple:

You have plaintext content, such as a message, document, or other data object, that you want to make unreadable to everyone other than selected recipients that you choose.

In order to do this, you need a key to jumble your plaintext into indetectable ciphertext. Depending on the type of encryption that you use, your recipients will need that same key or a different one to convert that ciphertext back to its original plaintext form.
Since keys are required both to encrypt and decrypt whatever content you're protecting, you must be careful to secure them just as you would the content itself.

**Sounds simple enough, right? Not exactly.**

If we're manually handing our car keys to a valet, that's an easy in-person exchange that doesn't pose many security risks. But most people and businesses don't communicate in-person as much as they do online. And unlike a car, digital information must remain readily accessible to the many people with whom it is shared. In order for that to effectively occur, encryption keys must be easily and safely distributable at scale.

What's more, the number of methods that we use to communicate online is constantly growing. Even though we create encrypted files on one storage application, we might also need to share those same files in an email attachment or using a different storage tool. Encryption keys don't always work when applied to different platforms, which means we often must manage multiple key exchanges for the same piece of data.
We account for these complexities like we do most other problems: we assume our technology providers will resolve them.

However, leaving key management blindly to third parties means your information is exposed and could be accessed without your knowledge or consent.

To ensure that your online data remains protected, it's critical to understand the different components of encryption key management, so that you know the right questions to ask when evaluating new and existing encryption technologies.
III. It All Starts with Encryption
Without encryption, there are no keys, so it’s important to know how modern encryption works before digging into the broader world of key management.

There are two common forms of encryption used today: symmetric encryption and asymmetric encryption. Some technologies employ combinations of both.

Symmetric key encryption uses the same key to encrypt and decrypt the data. A simple example is a password-protected PDF. The creator of that PDF uses a passcode to secure the document, while authorized recipients use that same passcode to view the PDF in plaintext form.

A more complex example of symmetric encryption can be seen on software developer platforms like Amazon Web Services and Cloudant. These platforms provide services that allow developers to encrypt their data and then return that same key to other users who need access to it.

Symmetric encryption can protect data at-rest, but it is not typically seen as an effective way to send encrypted data securely across different platforms. After all, how can the sender ensure that the key gets safely transmitted to the recipient?
Asymmetric encryption was created to address this concern. Asymmetric encryption uses two keys: one to encrypt data, and one to decrypt it. It’s often referred to as public key encryption, because people who use it make the encryption key public, but keep the decryption key private. Anyone can send out an email or file encrypted with the recipient’s public key, but only the recipient can read it, since only he has the private decryption key.

**Asymmetric Encryption**

Common forms of asymmetric encryption include: Pretty Good Privacy (PGP), Secure/Multipurpose Internet Mail Extensions (S/MIME), Transport Layer Security (TLS), and Secure Sockets Layer (SSL), each of which we will cover later.

While key management is tied to any type of encryption, it plays an even bigger role in asymmetric encryption, since the creation of multiple keys adds complexities that do not impact symmetric methods in the same way.
IV. The 5 Pillars of Key Management
As encryption technologies continue to evolve over the years, adoption is now growing faster than ever.

According to a 2016 Ponemon Institute survey, 41% of companies now encrypt their data. That 7% jump from 2015 is the largest in the survey’s 11-year history, which shouldn’t be surprising to anyone who reads the news.

The past two years have included an unprecedented number of high-profile breaches:

- **724k**
  - 724,000 personal accounts were affected by an IRS breach.

- **640k**
  - ADP breach exposed payroll data for over 640,000 companies.

- **117m**
  - 117 million LinkedIn email and encrypted password combinations were decrypted and illegally posted online.

- **20k/9k**
  - Hackers dumped the records of 20,000 FBI and 9,000 Department of Homeland Security workers.

- **20k**
  - More than 20,000 emails from Hillary Clinton’s campaign were leaked before the 2016 Presidential Election.

What makes these incidents all the more troubling is the abundance of hacks that came before them. Each of these organizations’ data was encrypted in some capacity, but it was still compromised.

**How is that possible?**
While encryption is a critical component of data security, it is only as effective as the methods that protect and distribute the keys being used.

Within the realm of key management, you should consider these five main areas as part of any comprehensive data security plan:

1. **Key Storage**

In the case of the missing Porsche 911, we learned that locking your car does not guarantee its safety. If the person holding your keys wants to steal your ride, he can still do so – unless other safeguards are in place.

The same principle applies to your data, which is why it's important to understand where the keys protecting your data are stored.

As a general principle, the person or company who stores your encrypted content should not also store the keys encrypting that content, unless you are comfortable with that third party accessing your unprotected plaintext.

Historically, people have had to sacrifice convenience for privacy in order to satisfy this principle.
Most common email and file sharing providers, such as Microsoft, Dropbox, Google, and others, provide some level of default encryption that is very easy-to-use, if not entirely automatic. But these providers often store all of the keys and content on their servers, which means that they could read your unencrypted data if motivated.

Asymmetric encryption technologies like PGP and S/MIME prevent third party access to unencrypted data by keeping encryption and decryption keys in the hands of separate parties. Since the recipient generates his private key and retains exclusive access of it, he does not have to trust a third party with storing or transmitting it, allowing him to retain sole access over the plaintext content:
PGP and S/MIME are both secure, but neither is very convenient. To start sending and receiving messages with either system, you need to install the software, generate a public and private key, register your public key, obtain your recipient’s public key, add it to your keyring (a directory of known keys), and encrypt your message – all before sending your message.

You also have to safeguard your own private key, since if anyone gains access to it, they’ll be able to read all of your email. Not to mention, you have to set up PGP or S/MIME and enter your key on every device you use, which wastes time and potentially exposes you to greater security risks.

2. Policy Management

While the primary role of encryption keys is to protect data, they can also add control capabilities to a given piece of content. Policy management is what allows an individual to add and adjust these capabilities.

For example, imagine if you could give your valet attendant a key that allowed him access to your car for 5 minutes (enough time to park your car), but that cut the ignition after that point (before he’d be able to drive off with it). Policy management occurs when you establish those restrictions.

Some encryption methods allow data owners to modify who can access the keys used to encrypt their data, which enables them to change an individual’s ability to decrypt content even after it’s been shared.

By setting policies on encryption keys, a user can revoke, expire, or prevent sharing of the keys, and thus of the unencrypted data, too. These policies can also audit when encryption keys were accessed, giving senders’ insight into when their encrypted content was read and by whom.
3. Authentication

Since keys enable people to unlock your encrypted data, it's important to verify your recipients' identities before giving them access to the keys.

Earlier, we alluded to the idea of allowing a stranger to park your car for you. That's what you're doing when you encrypt without authenticating who can access the keys. If you don't know who can access your encryption keys, you have no way of knowing who has access to your data.

In the case of parking a car, you use the valet attendant's uniform, name tag, and other social indicators to confirm that it's okay to give him your keys. By doing so, you're not preventing him from stealing your car, but you are indicating that you trust him to use it as intended.

That's what authentication is: verification that the person being given a decryption key should in fact be allowed to receive it. When encrypting digital content, there are several ways to achieve this.

Some tools, like the secure portals you may encounter when emailing with your doctor or logging into your bank, require you to create a unique username and password that the tool can use to verify your identity. Only once you've successfully logged in can you view the decrypted content.

Other encryption methods rely on your existing web credentials (such as those from a Google, Microsoft, or Facebook account) or physical verification devices (such as Common Access Cards and USB readers) to authenticate.

In some instances, as with PGP or S/MIME, the only way a recipient's identity can be verified is by manually exchanging encryption keys between both parties.
4. Authorization

Where authentication verifies who can access encryption keys, authorization verifies the actions that people can take on encrypted data once they've been authenticated.

It's the process that enforces encryption key policies and ensures that the encrypted content creator always has control of the data that's been shared.

Since encryption ties data to corresponding keys, some key management systems can assign different privileges to different recipients of the same data.

For example, you might share an encrypted file with two people, only one of whom you want to be able to print or download it, so you create key management policies to enforce these restrictions. Authorization is what ensures that these rules are passed onto your recipients when they try to access the encrypted file.

5. Key Transmission

Let's say you're using strong encryption. You store your keys securely, and you set policies on them to retain control after they've been shared. You've even implemented sophisticated authentication and authorization methods. Now what?

The final component of key management is also the most difficult one to satisfy. How do keys get transmitted to the people who need them, without giving access to the people who don't?

Like storage, key transmission usually requires people to choose between security and convenience.

Symmetric encryption enables easy encryption of data at rest, but it is not commonly used to protect data in-transit because of the risks involved in single-key exchanges. To encrypt an email using symmetric encryption, you have to contact your recipient and agree on a key, but if you send the key over the web, someone else could listen in and intercept it.

Asymmetric encryption methods like PGP and S/MIME do not require the sender to have access to the recipient's private key, which decreases the risk of this key being compromised during a transfer between parties.
However, PGP and S/MIME keys can only be shared with people who have configured the appropriate technology – which is not easy to do – and they can only be transmitted individually, which prevents sharing with groups.

The right key management framework enables key sharing that is both safe and easy. While there are a few solutions that claim to meet these criteria, most of them fall short in one of the areas we've discussed.

It's important to understand the common misconceptions out there before evaluating the right key setup for you or your organization.
V. Common Encryption Methods and their Key Misconceptions
Misconception #1: If vendors encrypt your data, this means they can’t access it.

There’s something troubling about most encryption features and solutions. Even though they promise to make your data unreadable to unauthorized parties, the vast majority of technology vendors still retain access to your unencrypted content.

Take TLS, for example. Unlike symmetric encryption, TLS provides an effective way to secure content as it travels from one point to another. That’s why 81.8% of cloud providers use TLS to encrypt customer data in-transit.

TLS provides an encrypted pipe through which data can be transmitted. Unlike PGP and S/MIME, TLS does not encrypt the actual content at rest. Instead, it ensures that unencrypted content is secure when travelling between locations.

As a result, third party providers typically have access to the unencrypted data that reaches their servers throughout this process:
SSL and TLS handle keys in the same way. When you send data using SSL or TLS, that data travels through a pipe that is encrypted with a private key that has been generated at the starting point in that transmission chain, and then with a public key that has been created for the endpoint in the chain. The data reaches the endpoint via an encrypted pipe, where it is then decrypted with the private key designated for that endpoint. The resulting ciphertext is then decrypted with a public key that was generated at the starting point in the chain. A Certificate Authority (CA) verifies that both parties are who they say they are (authentication) in order to prevent third parties from posing as authorized receivers and intercepting the message.

Think of an unlocked car that is transported from one public parking lot to another via a secure tunnel. While the car is travelling in that tunnel, you can guarantee its safety, but the tunnel only provides security during transmission. The parking lot owners, or any trespassers, can still get into the car when it’s not travelling.

SSL and TLS facilitate secure communications, but they do not encrypt the data itself. If a cloud provider stores your data in addition to transmitting it (which most providers do), these encryption methods alone cannot prevent that third party from accessing your data in unprotected form.

SSL and TLS facilitate secure communications, but they do not encrypt the data itself.
Portal-based encryption technologies suffer from the same limitations as SSL and TLS. With these tools, data is stored on a portal that is hosted on the vendor's server. Recipients establish user IDs and passwords to authenticate themselves and access their decrypted content within the portal application.

This architecture adds additional security by requiring new usernames and passwords to access sensitive content. However, most portal providers send unencrypted plaintext over SSL connections and manage the encryption keys that protect this data. So, even once that data reaches the secure portal, the vendor has control over both the keys and the content needed to access the customer's unencrypted data.
As we discussed earlier, the best way to prevent technology vendors from accessing your plaintext data is to separate where keys and content are stored.

Certain products – such as Box KeySafe, Intralinks CMK, and SafeNet KeySecure, among others – attempt to do so by re-encrypting keys and storing them on the customer's server. When you use KeySafe, for example, Box generates an encryption key that it uses to secure any files that you upload to Box's servers. Box then sends the encrypted file to the customer's servers, where customer-hosted encryption keys rewrap the ciphertext.

While solutions like these give customers exclusive access to some encryption keys, these keys secure data after it has already been accessed by the vendor in plaintext form. What's more, even though the vendor encrypts files upon upload to its servers, it also generates the keys used to perform this encryption, which means that it handles unprotected data protection keys as well as unprotected files.

This combination gives encryption providers similar control over content as most other cloud providers. Customer-managed key solutions enable customers to control or host some of their encryption keys, but they don't do so in ways that ensure full privacy and control.

Misconception #2: If you encrypt, hackers cannot get access to your data.

“Okay,” some people might say. “I'm fine with vendors having access to my data, as long as hackers won't be able to get to it.”

Unfortunately for these folks, it’s virtually impossible to guarantee that.

However, certain encryption methods make it easier for hackers to gain access to sensitive data than others.

We spoke about the pitfalls of TLS before, but they span beyond the technology’s inability to secure data at rest. When an email is encrypted with TLS, it only stays encrypted if the recipient’s email server can receive TLS-encrypted messages. If the server only has an old, weak version of TLS, or doesn't support it at all, the email will not be encrypted once it leaves the sender's domain.

As a result, the otherwise encrypted TLS pipe becomes exposed, preventing data from traveling securely. Hackers can also compromise TLS by gaining control of a server, or by using a fake security certificate to pass a scam website off as the real thing.
While TLS is not known as a good means for at rest protection, full-disk encryption (FDE) is. That’s why most laptops and smartphones come equipped with this capability.

FDE uses symmetric keys to make a device’s hard drive data unreadable to unauthorized parties. Keys are loaded onto the hard drive, which must be turned on in order for decryption to take place. Most devices gate hard drive access with a login passcode as an additional layer of defense. If someone stole your laptop or smartphone and was unable to log into it, FDE would keep the hard drive protected.

The term “full disk” implies complete and persistent security, but there are significant limitations.

For starters, most devices do not enable FDE by default. Unless you turn the feature on, your hard drive will be vulnerable to attacks, even when the device is powered off.

Also, while some devices (like the iPhone) enforce login passcodes by default, many don’t. Without a passcode, your hard drive data will become unprotected the moment the device turns on, since FDE stores encryption keys on the device itself and only encrypts when it is turned off.

Additionally, FDE will not keep malware from infecting your device, and it will not prevent remote attacks. In order to protect your organization from these types of breaches, you must encrypt the actual data in addition to the drives and servers hosting it.

That’s where Virtru can help.
VI. Introducing Virtru
Client-Side Encryption
Compared with other encryption approaches, Virtru provides the best of both security and ease of use.

Public key encryption methods like PGP and S/MIME separate keys from encrypted content and restrict third party access to unencrypted data. However, they are difficult to implement and require advanced coordination with recipients.

Portal-based systems have been around for decades, but they don't encrypt the full path between the sender and recipient, leaving data more vulnerable than most users expect. TLS is easy to enable, but it only protects data as it travels, and it only works when both parties' email servers have TLS configured.

Virtru offers multiple key management options to enable easy-to-use email and file encryption that protects data no matter where it travels – and prevents third parties from ever accessing unencrypted content.

<table>
<thead>
<tr>
<th>Key Storage</th>
<th>Stored separate from encrypted content; symmetric keys hosted on Virtru's AWS; additional asymmetric keys can be hosted exclusively on customer premises.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy Management</td>
<td>Set expiration, disable forwarding, and watermark PDFs upon encryption; revoke and view read receipts for shared content.</td>
</tr>
<tr>
<td>Authentication</td>
<td>OAuth, SAML, OpenID, or email verification link.</td>
</tr>
<tr>
<td>Authorization</td>
<td>Managed exclusively by Virtru Access Control Manager.</td>
</tr>
<tr>
<td>Key Transmission</td>
<td>Keys transmitted separate from encrypted content; shared via TLS between email clients and servers.</td>
</tr>
</tbody>
</table>
To ensure a seamless experience, Virtru performs encryption directly within existing email and file sharing platforms, such as Gmail, Google Drive, Microsoft Outlook, iOS, Android, and others. Virtru’s apps and plugins hook into these tools to encrypt data on the client-side, before it ever leaves the device.

When you encrypt an email with Virtru’s Software-as-a-Service (SaaS) product, a symmetric key encrypts the message the moment you begin composing it from your email client. When you hit ‘send,’ the symmetric message key travels via TLS from the client to Virtru’s AWS server, while the encrypted email is sent out and stored by your email provider – just like a regular email would be.
Once the encrypted email reaches a given recipient’s email client, the recipient must authenticate himself using either his existing email credentials (via an open verification standard called OAuth) or a message verification link. Post-authentication, Virtru sends the symmetric key the receiving email client, again using TLS. If the recipient has Virtru installed, his email client can use the message key to decrypt the email directly within the inbox.

If the recipient has Virtru installed, his email client can use the message key to decrypt the email directly within the inbox. If the recipient does not have Virtru installed, he can view the unencrypted email in Virtru’s Secure Reader, a browser-based web client that performs decryption without requiring recipients to create new login credentials or install any software.
By separating the storage location of encryption keys and content, Virtru's encryption prevents providers from getting access to decrypted data. While Virtru stores the symmetric message keys, they cannot decrypt content with these keys, since that data is stored and transmitted by the email provider. And even though the email provider has access to the user's content, this content is encrypted, and they do not have access to the encryption keys needed to decrypt.

This split-knowledge architecture is critical for organizations looking to comply with Health Insurance Portability and Accountability Act (HIPAA), Criminal Justice Information Systems (CJIS), or International Traffic in Arms Regulations (ITAR) requirements that restrict third party access to sensitive data.

As we discussed, the safest way to manage encryption keys is to host them yourself, and Virtru provides that option for organizations via its Customer Key Server (CKS) capability.

The CKS is a physical device or cloud server that the customer hosts entirely on their premises in the container-storing platform of their choice. The Virtru CKS adds asymmetric encryption to Virtru's pure SaaS offering to give the customer complete and exclusive access to the keys encrypting their data.

The Virtru CKS also provides enhanced control for users via advanced policy management. Even though Virtru hosts the encryption keys protecting your data, you control who can access these keys, thus enabling you to:

- Revoke any email, even after it's been read
- Set expiration dates on emails
- Track where emails are forwarded, or disable forwarding altogether
- View read receipts to see when recipients have read your emails
- Watermark PDF attachments for individual recipients
When you encrypt an email under this model, your Virtru email generates a message key that is encrypted with a CKS public key. The CKS hosts the private key needed to decrypt this public key and unwrap the message key, but only you can access it, since the CKS is hosted on your premises. Virtru’s servers only store encrypted keys, so they never have access to decrypted message keys.

Receiving Virtru clients – either Virtru’s Secure Reader or an inbox that has a Virtru plugin installed – also have public/private key pairs. The CKS rewraps message keys with the receiving client’s public key before it is transmitted to Virtru’s servers and eventually to the receiving client itself. The receiving client, which sits on the recipient’s premises, contains the private key needed to unlock the rewrapped message key and finally decrypt the message.

You should consider the Virtru CKS if you’re looking to:

- Enable easy-to-use client-side email encryption without having to trust third parties with encryption keys or unencrypted content.
- Ensure that you are the only entity that can respond to government access requests and subpoenas.
- Meet data residency requirements by specifying the locations where your encryption keys are stored.
- Destroy encryption keys to make emails permanently unreadable.
Prior to the Virtru CKS, most existing customer-held key solutions required organizations to trust their cloud provider with the data protection keys directly protecting their content. Because they can access those keys, cloud providers also have access to the underlying plaintext content with these solutions. Virtru is the first zero-knowledge key distribution service in which no third party can ever access unprotected content or the data protection keys.

If you need even more security, you can back up the Virtru CKS with a Hardware Security Module (HSM), such as the SafeNet Luna. The HSM is a physical device hosted on-premise by the customer to add an additional layer of encryption on top of the CKS. With the HSM, customers can provide their own public/private key pairs and share them across any Virtru clients. This is a particularly attractive model for anyone looking to use PGP or S/MIME without having to handle key exchanges manually or require their recipients to install anything.
### What Keys and Content Can Cloud Providers Access?

<table>
<thead>
<tr>
<th>Key Management Setup</th>
<th>Data Protection Keys</th>
<th>Plaintext Content</th>
<th>Customer-Held Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Customer Managed Key Solutions*</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Virtru SaaS</td>
<td>YES</td>
<td>NO</td>
<td>N/A</td>
</tr>
<tr>
<td>Virtru CKS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

*Such as Google Cloud KMS, Box KeySafe, Intralinks CMK, and SafeNet KeySecure*
With each key management configuration, administrators can use Virtru to monitor data going in and out of their domain from a centralized dashboard. Because customers always manage their encryption keys with Virtru (even with pure SaaS key management), they can view audit trails of when keys have been accessed, and thus see when emails have been read by whom.
Whether you need to meet regulatory compliance, protect corporate info, or simply prevent third parties from accessing your content, Virtru’s three key management options provide a secure, easy-to-use data protection foundation for any individual or organization.

More Information on Encryption Key Management

As thought leaders in encryption key management, Virtru offers a valuable array of resources to help you navigate the often murky data security landscape:

**Blog Post:** [How Encryption Key Management Stops Hackers](#)

**Blog Post:** [Virtru Launches Hardware-Backed Encryption Key Management](#)

**PDF:** [Virtru Key Management One-Pager](#)

**Video:** [Virtru Encryption Key Management](#)

**Webinar:** [Secure Your Org: A Practical Guide and Case Study on Email Encryption](#)

**Case Study:** [Virtru Enables Cloud Migration for More Than 15k State of MD Employees](#)
### Encryption Key Management Needs Assessment Checklist

The following checklist will help you evaluate your organization’s encryption key management needs and determine appropriate solutions to meet your requirements:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Yes or No?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does your organization use any type of email or file encryption?</td>
<td></td>
</tr>
<tr>
<td>Does your encryption protect emails and files at rest and in-transit?</td>
<td></td>
</tr>
<tr>
<td>Do personnel use client-side encryption to share sensitive data?</td>
<td></td>
</tr>
<tr>
<td>Does your encryption provider ever have access to encrypted data?</td>
<td></td>
</tr>
<tr>
<td>Does your encryption provider ever have access to encryption keys?</td>
<td></td>
</tr>
<tr>
<td>Does your organization share any data that you don’t want your email or file sharing provider to access?</td>
<td></td>
</tr>
<tr>
<td>Do you trust your technology providers with your sensitive data?</td>
<td></td>
</tr>
<tr>
<td>Do you want the ability to directly respond to government surveillance requests for your organization's data?</td>
<td></td>
</tr>
<tr>
<td>Does your organization have any products that are competitive with any of your technology providers’ products (i.e., Microsoft, Google, Amazon, etc.)?</td>
<td></td>
</tr>
<tr>
<td>Does your organization face any data residency requirements?</td>
<td></td>
</tr>
<tr>
<td>If yes, does your cloud provider guarantee that your organization's email and file data will not leave your premises?</td>
<td></td>
</tr>
</tbody>
</table>
**Enhance Your Email and File Security Today**

Whether you are already using a cloud service, like Google or Microsoft, or are looking for stronger on-premise security, [Virtru’s client-side email and file encryption](#) with customer-hosted keys is the easiest, most secure way to comply with privacy and compliance standards. To see if Virtru is right for your organization, you can:

- [Download Virtru for free](#) now and start sending securely
- Contact us to [request a demo](#)
About Virtru

Virtru believes that personal privacy depends on the businesses, governments, and institutions that hold our information. Virtru products make it easy for businesses and individuals to maintain control over access to their emails, documents, and other data – regardless of where they've been shared. More than 5,000 organizations around the world rely on Virtru to secure their most sensitive information.

www.virtru.com
sales@virtru.com
Contact us for more information

www.virtru.com