



## Protecting Digital Certificates and Access Keys with Keeper

## How Can I Store and Protect my Digital Certificates and Access Keys?

---

Out of the box, both Keeper's consumer and business versions support the storage and protection of digital certificates and keys. Encryption keys and digital certificates provide a critical security layer that protects every digital asset in an organization. According to a recent [Ponemon study](#), breaches due to trust-based attacks are caused by the mismanagement of digital certificates. A successful attack carried out against a digital certificate can have disastrous effects on an organization. On top of the added security threat, expired certificates [cost companies millions of dollars](#) in lost business. Start your [free trial of Keeper Enterprise](#) today to protect your digital certificates and keys.

## What are Trust-Based Attacks?

---

Digital certificates and encryption keys are essential to business trust. They secure data, keep communications private and safe, and establish trust between communicating parties. Despite their importance, many businesses leave their organizations vulnerable to compromise and breach by allowing the management of certificates and keys to be viewed as an operational problem, instead of a security vulnerability that needs to be rectified immediately.

Due to this perceived vulnerability, hackers are increasingly focusing on keys and certificates as an attack vector. Malicious actors use stolen keys and certificates to gain trusted status and then use that status to evade detection and bypass security controls. Criminals use trust-based attacks to infiltrate enterprises, steal valuable information and manipulate domains. If private keys used to sign a digital certificate get in the wrong hands, the system can be breached and the website can be overthrown. If private keys are lost, then significant time and energy will likely be wasted trying to access systems or renew certificates. If code signing certificates used to sign an iPhone or Android app are compromised, a rogue developer could launch malware using the breached corporate identity.

To mitigate trust-based attacks, certificates and encryption keys need to be safely protected and stored securely to prevent them from being misplaced or falling into the wrong hands.

## The New Challenges Facing Sys Admins

---

Developers and IT Admins are constantly plagued with managing new keys and certificates that are about to expire or need to be rotated. Over time, the number of keys and other developer-centric digital certificates grows rapidly. Multiply these by the number of environments SysAdmins need to control (dev, sandbox, staging, production), and now you have four times the number of keys to manage. On top of managing all of these keys and certificates, Sys Admins are having their responsibilities increased by other factors, including:

### > Remote Access

As systems become hardened, network admins add additional layers of complexity to remotely access systems and networks. Multiple layers of SSH, port forwarding, certificates for VPN authentication, multi-factor authentication, X.509 certificates, RSA private keys, etc. SSH keys for accessing systems have to be managed and tracked by someone, and all of those keys need to be expired and rotated.

### > Protecting Cloud Services

As organizations and individual developers make use of cloud services such as Amazon AWS, Google Cloud, and Azure, they are no longer responsible for managing just usernames and passwords. Now they are now faced with storing and protecting Access Keys, Secret Keys, and API Keys. These keys are even more critical to protect because they can be used to directly access and control cloud-based services.

### > Managing the Migration to HTTPS

With the push towards optimal SEO and end-user protection, Google is pushing all website owners to migrate towards using HTTPS/SSL. This requires the generation of private keys and digital certificates for every domain name that is accessed by users on a web browser.

### > Deploying Apps

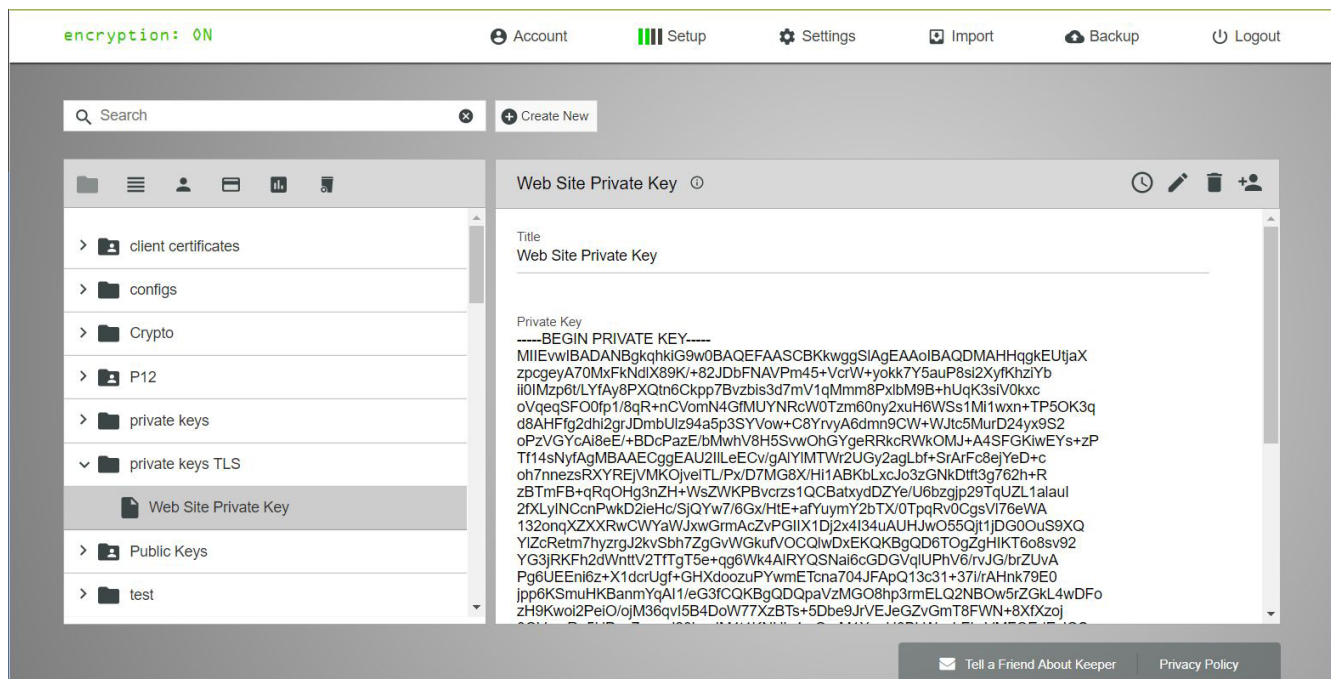
Software developers are faced with creating certificates and private keys to digitally sign their software applications. Apple, Google, and Microsoft all require the use of code signing certificates to distribute applications through their platforms. Each individual team member within a software company must be responsible for managing their own keys and ensuring that production-level keys are protected.

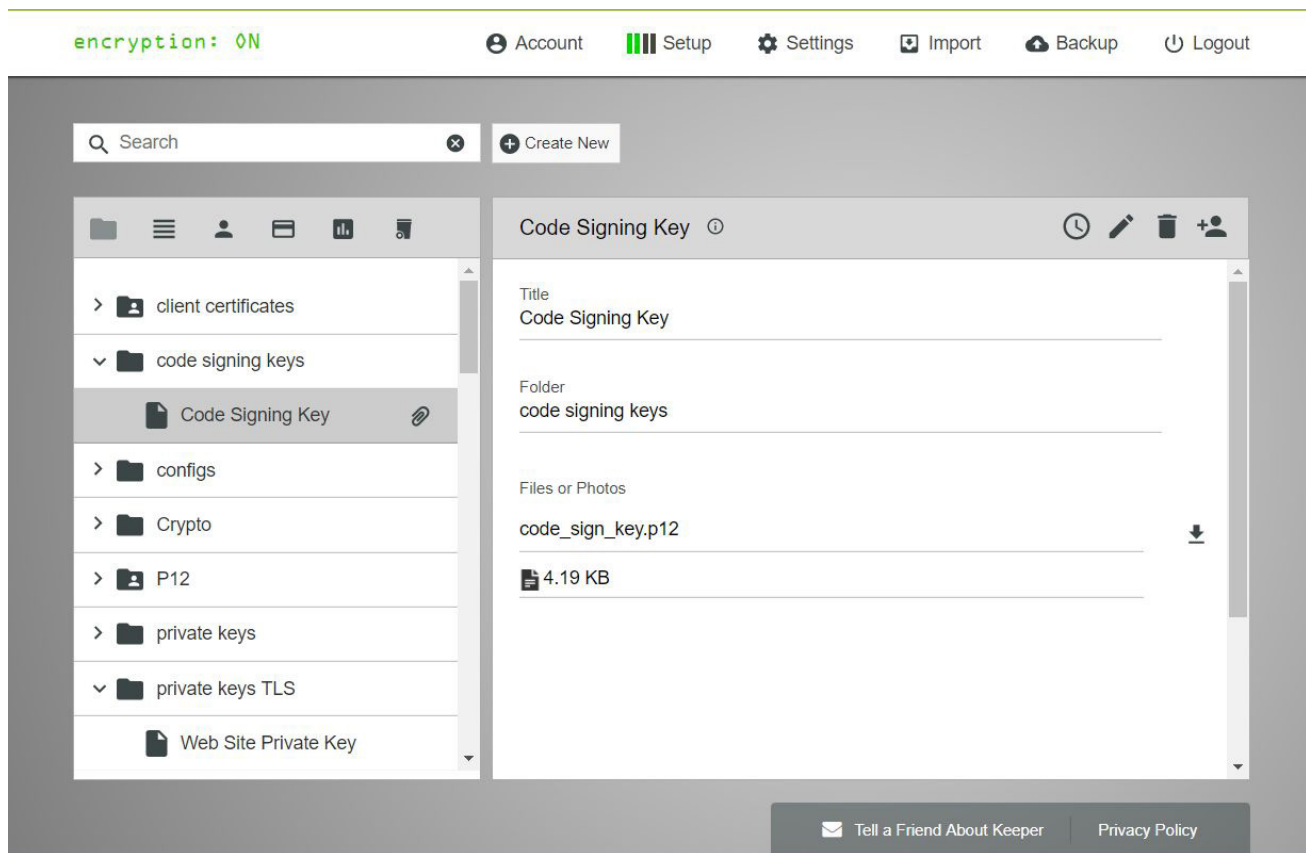
## Where should you manage certificates and keys?

All of these certificates and keys have to be protected somewhere safe. The worst thing that IT admins or developers can do is store certificates and keys in plaintext (or even worse, in a Github repo). SSH Keys stored on end-user's computers are goldmines for malicious intruders and malware specifically designed to escalate their privileges. So where is the safest place to store certificates and keys?

### Simple answer: store them in Keeper.

Keeper stores all of your private keys, digital certificates, access keys, API keys and other secret data in an encrypted digital vault. Keeper provides a simple way to access your private info across any device type or OS. With Keeper, these digital assets are fully encrypted locally on your device with 256-bit AES and the ciphertext is stored in Keeper's Cloud Security Vault.



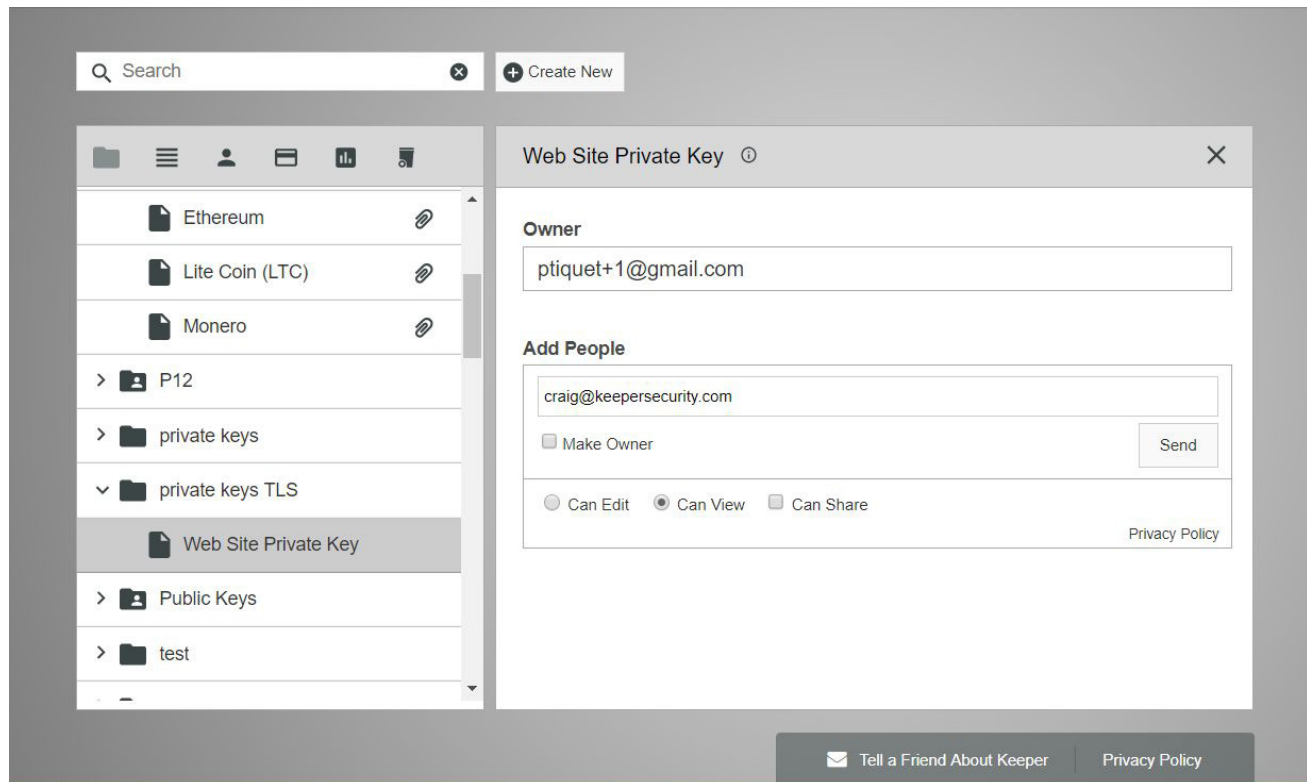


The benefits of storing digital certs and keys in Keeper are many:

1. Zero-knowledge architecture – only YOU have access to decrypt your files.
2. 256-bit AES protection with record-level encryption keys.
3. Accessibility across computer, web browser, and mobile devices.
4. Synchronization and backups across all of your devices and computers.
5. Easy, secure record sharing (or record transfer to another privileged user in case of emergency).
6. Access to the Keeper Commander SDK from any operating system.
7. Support for the storage of encrypted or binary-encoded keys or certificates.

## Secure Sharing

One of the best features of Keeper is the built-in secure sharing mechanism. You have the ability to customize Keeper to meet the needs of your company and organization structure. Privileged users can be added to any Keeper record, with different levels of permission. For example, the developers might only need access to the sandbox level keys, and the deployment manager or team lead may need access to the production keys.

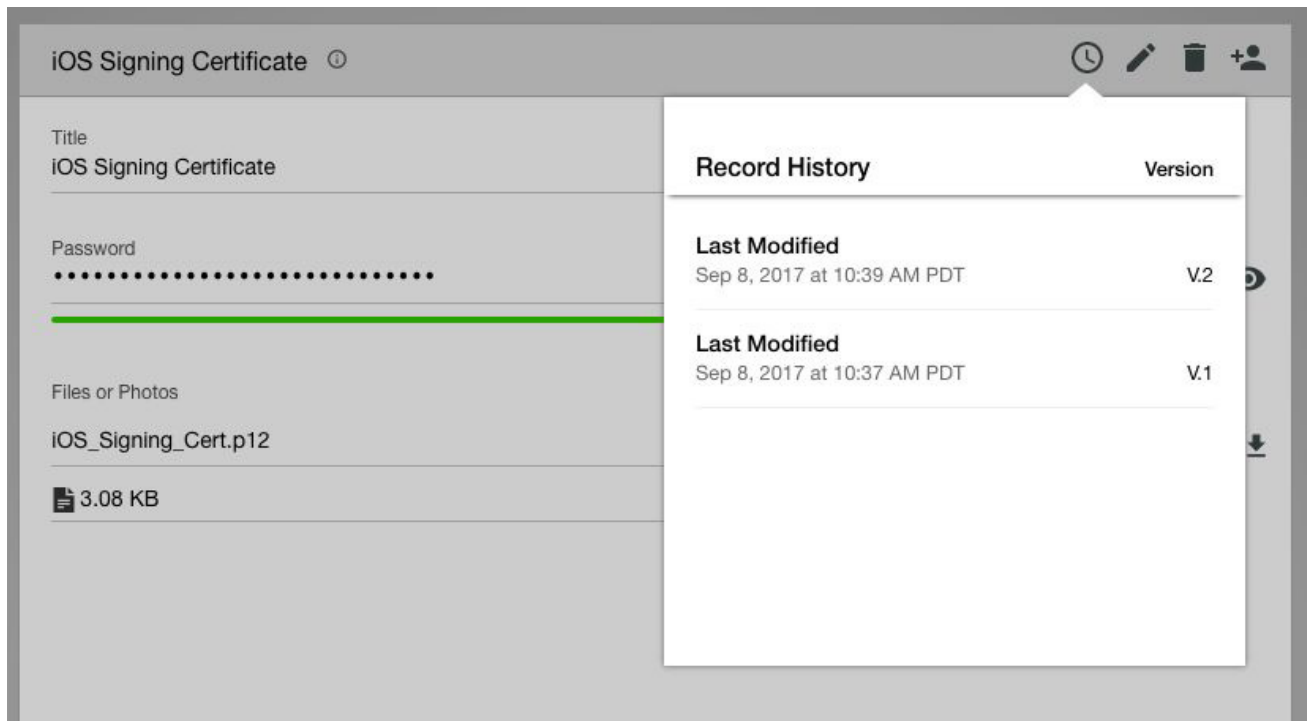


## Easy Ownership Transfer

Keeper allows you to easily and securely change record ownership. For example, the IT admin can generate an AWS Access Key for another team member and simply click “make owner” inside Keeper’s sharing screen. The record is encrypted with the recipient’s public key, and the recipient decrypts the record with their RSA private key. Then, the ownership of the record is transferred and it appears in the recipient’s vault. The beauty of this model is that data is never stored, transmitted or leaked in plaintext. The communications channel used by the team members is fully encrypted during the entire process.

## Track Audit Changes with Record History

Keeper provides a full version history of every record stored in Keeper. What if you executed one of the 25 commands wrong and saved the wrong certificate or private key in your vault? No problem! Just click on the “Record History” button and revert to the previous version. Every version is stored in Keeper, fully encrypted. If you delete a record by accident you can just click on the trash can and restore the record.

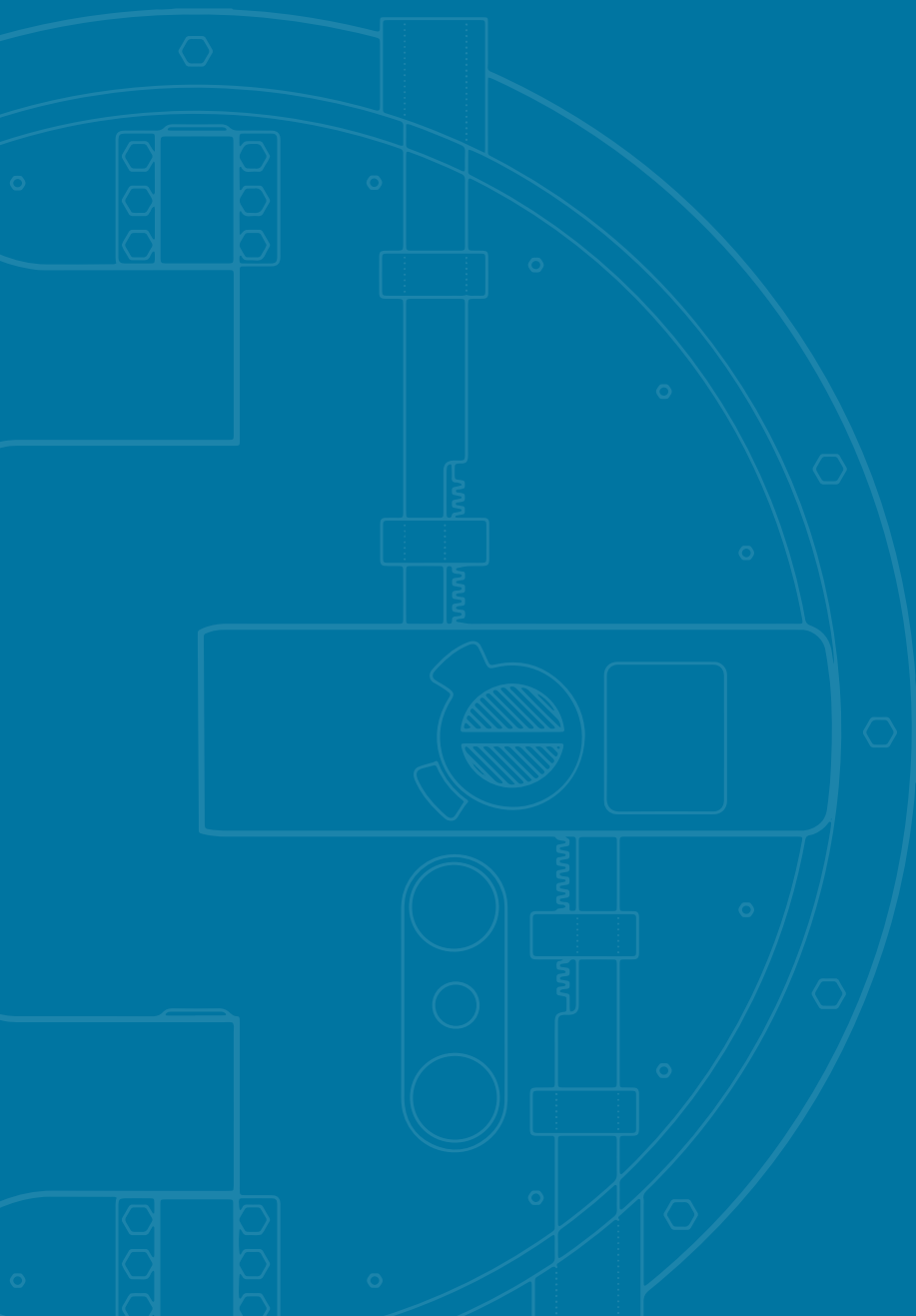


## Getting Started with Keeper

Begin your [free trial of Keeper for Business](#) today and start securely storing your certificates and keys while using Keeper's enterprise-strength password manager and digital vault to protect your company and streamline your business processes.



Protecting Digital Certificates and Access Keys with Keeper



## Contact

 [keepersecurity.com](https://keepersecurity.com)

 312.829.2680

 [sales@keepersecurity.com](mailto:sales@keepersecurity.com)