



BUYER'S GUIDE

A detailed image of an eagle in flight, with its wings spread wide, is positioned behind the main title text. The eagle is dark brown with lighter feathers on its wings and tail. The background is a soft-focus image of a blue sky with wispy white clouds.

# Zero Trust proof Enterprise SSH Key Management (EKM)

# SSH keys are like passwords – except too often unmanaged

The enterprise IT (Information Technology) landscape is increasingly hybrid and extended. Data and applications no longer reside only in data centers but are distributed across multiple cloud platforms. That data is accessed from different geographical locations by in-house and outsourced power users alike. What's more, in many enterprise environments, most of the connections are automated between servers.

Secure Shell (SSH) is still the de facto method for Linux, database & network admins, and application support teams to securely connect to servers and applications within them. That access is made by using SSH keys. Just like passwords, SSH keys are critical access credentials that provide access to critical data, for example:

- bank transfers
- credit card and medical data
- email servers
- firewalls and VPNs (Virtual Private Network)
- company internal networks

But unlike passwords, SSH keys are typically not managed, even though they outnumber passwords 10 to 1 in large enterprises. Most companies have Privileged Access Management (PAM) software that vault passwords – and even keys. But even in the best-case scenario, PAMs only vault 20% of all SSH keys in distributed architectures. What's worse, without proper SSH Key Management, SSH Keys can also be used to bypass those expensive legacy password PAMs, reducing your security posture dramatically.

Securing and managing SSH encryption keys is just as critical as managing passwords in business-critical environments.

# Shortcomings of existing key management solutions

There are plenty of solutions that attempt to simplify and manage SSH key lifecycle management. However, only a few of them are equipped to handle the complexities of modern and distributed digital environments with key estates encompassing hundreds of thousands of encryption keys and comprise dynamic cloud services.

Lets' look at the shortcomings of some of the methods.

## **A) Vaulting Keys in PAM**

The challenge with vaulting keys is changing all automation scripts to learn to retrieve a Key from the Vault for each transaction. In large enterprise contexts, we are talking about hundreds of thousands of configurations. Also, if the applications' authentication to a target application or a database is always dependent on key retrieval, the PAM becomes a single point of failure for critical transactions.

## **B) Keys be saved in a directory (LDAP, AD) and retrieved as necessary.**

Directory services are not secure enough and are partially vulnerable to bots. Microsoft recommendation for Active Directory to segregate and forest ADs to protect e.g. Operational Technology (OT) assets in manufacturing segments.

## **C) A holistic PKI architecture built around keys**

A public key infrastructure (PKI) is a set of roles, policies and procedures to manage public-key encryption. It is possible to build key management around permanent X.509 Certificates with:

- i) Root certificate authority (CA)
- ii) Intermediate CA
- iii) Issuing CA
- iv) Public Key
- v) Private Key
- vi) Certificate Store
- vii) Certificate Revocation list (CRL)
- viii) Delta CRLs
- ix) Certificate Management (enrollment / issuance / validity / revocation / renewal)
- x) Certificate Policy (CP)
- xi) Certificate Policy Statement (CPS).

This is exactly as complicated as it looks and is duplicate work, since SSH keys already have private-public key authentication and encryption built-in. You also need additional certificate management software that adds to complexity. You also need to remember to renew those certificates before their "best before date".

# Enterprise SSH Key Management – What to consider?

Since SSH keys can be managed in various different ways and by all types of tools, it's important to set the proper criteria for the solution. When evaluating an Enterprise SSH Key Management solution, it is important to pay attention to the following key capabilities.

- Can you centralize key management?
- Is the solution comprehensive enough?
- Can you delete keys without breaking critical connections?
- Can you identify policy and regulation violating keys with ease?
- Can you automate manual tasks?
- Does the solution reduce the key management overhead?

Let's explore some of the most common challenges in detail.

## 1 Discovering all keys and centralizing their management

**A single key can make you fail your IT audit.**

Yet, the most difficult part of enterprise key management is discovering all keys and managing them under a single pane of glass. This is because by default, the SSH key distribution model is decentralized with no clear ownership. SSH keys are also easy to self-provision, they are distributed across critical infrastructures, and they never expire, so their numbers quickly accumulate.

Moreover, keys are hidden behind jump servers and used by automated machine-to-machine (M2M) connections. As an example, any security tools like Privileged Access Management (PAM) claim to manage keys. But even in the best-case scenario, they only vault 20% of all SSH keys in distributed architectures.

Centralized visibility and control into the key estate at scale helps businesses to locate vulnerable, rogue, ungoverned, uncompliant, and policy-violating keys – and then act accordingly in a systematic fashion. Moreover, finding other critical SSH components like SSH login or SSH configuration files takes a dedicated solution purpose-built for this task.

## 2

### Replacing vulnerable, ungoverned, and policy-violating keys without disruptions

Sometimes enterprises know they have outdated and policy-violating keys in their environment. They are still often faced with two major challenges. The first one is a surefire method of eliminating, upgrading, or replacing encryption keys without interrupting or even disrupting a critical connection.

The second one is the complexity of the key estate: the sheer number of keys to be managed and their tendency to build one-to-one as well as one-to-many connections.

An enterprise-grade key management solution should be able to remediate, delete, recover, and rotate keys without causing disruptions to connections or leading to operational downtime. It should also build a 'web of trust' between all connections so that you can manage their use with a high-level of confidence and know exactly how trust relationships are built between servers.

Many organizations have documented proper use of SSH keys and educated their staff about the compliant ways of using them. But even with the best guidance, people forget recommendations and third parties are often not properly apprised of the policies.

Flagging and stopping the use of policy-violating keys and mapping your entire key estate against existing regulations (like GDPR, HIPAA, SOX [Sarbanes Oxley]) through software is a much more effective way of staying compliant.

It ensures that policies and regulations – and the restrictions that come along with them – are automatically built into your process when keys are used, instead of being a separate part of it.

## 3

### Enforcing policies, alerts on violations and reports on the state of your keys

## 4

### Non-intrusive deployment without changes to IT architecture

Many key management solutions or tools require changes to your existing configurations, need rewriting applications or force you to onboard keys to a vault before they are operational. This makes both the deployment and maintenance of the solution challenging, especially in vast key environments with dynamic cloud applications and when working with outsourced workforce with temporary access needs.

It is better to manage SSH encryption keys in a native way: leave them where they are but prevent their ungoverned use. The important thing is to ensure that the keys are disarmed so that they cannot be used outside the centralized key management solution.

This non-intrusive deployment model ensures that the solution can be implemented without long downtime to your operations due to intensive changes to your architecture or scripts.

## 5 Automating the SSH lifecycle to simplify the effort of staying compliant

An SSH key management solution should reduce manual tasks from the key management lifecycle and automate operational tasks. Examples include:

- Integrating key authorization processes with existing ticketing systems for fluid workflow approval and delegation of responsibilities between application owners
- Automating key provisioning, rotation, and remediation
- Automatic detection and prevention of policy violations
- Configuration lockdown
- Compliance process enforcement
- Full audit trail of activities with optional session recordings

Automated tasks reduce the risk of human error and save time and resources, especially with tasks that are repeated often. Together with one customer, we calculated that the savings they gained reached millions per year because of radical reductions in time, effort, configurations, maintenance, and manual work.

## 6 Reducing key management overhead with just-in-time (JIT) Zero Trust access – without keys

Taking control of all SSH keys is the first step in mitigating SSH key risks. The next step is to reduce management overhead associated with rotating, deleting, and issuing thousands of keys per month.

With an advanced SSH Key Management solution, access is granted on-demand, just-in-time, and without permanent keys. This 'keyless' access paradigm allows privileged users to access their targets without leaving any keys behind to be managed at all and improves security by reducing the number of credentials to secure.

What's more, you can onboard your existing keys to a keyless access model, which allows you to reduce the number of keys you need to manage by the thousands. It also enhances key management with Zero Trust, just-in-time and just enough access (JEA) models, since every session is verified every time it is established with the right level of privilege – without granting always-on authorization to anyone.



# Universal SSH Key Manager<sup>®</sup> for Zero Trust Enterprise Key Management

UKM Zero Trust is an Enterprise Key Management solution that automates governing keys according to compliance standards and security policies to mitigate risks, reduce key management overhead, and help pass IT audits.

## Centralize and automate SSH key lifecycle management at enterprise scale

UKM Zero Trust automates key provisioning, rotation, and remediation and integrates key authorization processes with ticketing systems for delegation of key ownership and workflow approvals. It also automatically detects and prevents policy violations, uncompliant keys, and rogue keys to help you pass IT audits and mitigate risks.

With UKM Zero Trust, you can achieve a configuration lockdown that prevents the unauthorized use of SSH keys outside the solution. It is your security gateway to your SSH environment.

## Ensure you have all keys under control

UKM Zero Trust is the most comprehensive Enterprise SSH Key Management solution on the market. It discovers the most hard-to-find keys both on the clients and the servers, locates SSH configuration and login files, and puts keys hidden behind jump servers or hidden folders under management.

With the solution, you:

- Prevent leaving keys in the hands of third parties when their contract expires
- Stop access from test to production
- End violations to Segregation of Duties (SoD)
- Delegate keys to application owners and teams securely
- Demonstrate full compliance in audits

## Track, audit, and log interactive and automated SSH sessions

UKM Zero Trust can track, log, and audit both interactive and automated SSH connections alike. Based on our experience, 80% of all SSH connections are run automated between servers. UKM Zero Trust assigns an identity to every user and machine session, making unidentified sessions impossible.

## Reduce key management overhead and complexity with just-in-time (JIT) Zero Trust access

UKM Zero Trust paves the way for a keyless future. In this model, the SSH connection is established with short-lived certificates that are created just-in-time and contain the keys needed to make the connection. The certificates then expire automatically after authentication, leaving no keys behind to manage, share, or lose. This process repeats itself for the next connection or session automatically again.

Migrating to keyless SSH connections radically reduces the complexity and management overhead of keys. The less keys you need to manage, the fewer points of trust you have in your network, making it a great fit for a Zero Trust framework.

With the solution's built-in hybrid model, you still manage the keys you need to but migrate to a keyless paradigm at your own pace.

## Leverage the expertise from the inventors of the SSH protocol

We at SSH are the inventors of the Secure Shell protocol and have 25 years of experience with keys and solving SSH-related challenges. We also introduced the first, enterprise scale Key Management solution that manages keys in the native way. Then we introduced the Zero Trust Key Management solution that is based on the patented, ephemeral certificate based technology.

Our experts have been building solutions and solving SSH-related challenges together with the most demanding Fortune 500 companies for decades.

## Zero Trust proof solution for Enterprise SSH Key Management

Built together with Fortune 500 customers, UKM Zero Trust is your trusted workhorse for automated discovery, visibility into security standards, and centralized management of SSH keys across hybrid multi-cloud environments. Manage thousands of keys with the least effort and risk.

[Learn more about UKM Zero Trust at SSH.COM.](https://ssh.com)





## **Finland**

**SSH Communication Security Oyj**  
Karvaamokuja 2 B 00380 Helsinki  
[www.ssh.com](http://www.ssh.com)  
**+358 20 500 7000**  
[info.fi@ssh.com](mailto:info.fi@ssh.com)

## **USA**

**SSH Communication Security, INC.**  
434 W 33rd Street, Suite 842  
New York, NY, 10001, USA  
[www.ssh.com](http://www.ssh.com)  
**+1 781 247 2100**  
[info.fi@ssh.com](mailto:info.fi@ssh.com)

## **Hong Kong**

**SSH Communication Security LTD.**  
35/F Central Plaza, 18 Harbour Road  
Wan Chai  
Hong Kong  
[www.ssh.com](http://www.ssh.com)  
**+852 2593 1182**  
[info.fi@ssh.com](mailto:info.fi@ssh.com)