Challenges of Key Management

By Dr. Malini Bhandaru, Intel Software Architect

Effective, efficient management of cryptographic keys is vital to the operation of secure infrastructure. Developing a mechanism to support key generation, storage, access, exchange, and availability is a complex undertaking that must address a number of challenges.

For example, generating high-quality keys requires a rich entropy source. The keys themselves must be encrypted, to keep them secret both in storage and in transit. Moreover, the loss or destruction of a key often causes the unrecoverable loss of the data it protects, making fault tolerance and secure backup critical to the design of key-management systems. The mechanism must also scale effectively.

Robust key-access controls are required, both to authenticate users and to restrict their access to only those keys for which they are authorized. Other requirements that add to the complexity of developing and deploying a key manager include the following:

- **Administration functionality,** such as limiting the number of times a key is used, retiring expired keys, changing a master key, and auditing key usage and life cycle events
- **Support for various types of keys,** including interaction by public-private key pairs with a certificate authority for certification
- **Accommodation of multiple formats** to facilitate key exchange between diverse key repositories

OpenStack Operation and the Need for Encryption

OpenStack is an open-source cloud platform for Infrastructure-as-a-Service (IAAS). Since its initiation in 2010, the OpenStack project has grown rapidly, with hundreds of companies investing and becoming involved in the community. The environment's modular architecture is based on a collection of services—each of which is generally referred to by its code name—that include the following:

- **Authorization and access control** (Keystone)
- **Virtual-machine scheduling** (Nova)
- **Object storage** (Swift)
- **Block storage** (Cinder)
- **Image storage** (Glance)
- **Network and firewall provisioning** (Neutron)
- **Usage-data collection** (Ceilometer)
- **Web-dashboard capabilities** (Horizon)

Through Horizon, a registered cloud user logs into the OpenStack platform, and his or her credentials are authenticated by Keystone. The user then selects a virtual machine (VM) image stored in Glance, specifies physical host machine preferences (such as the number of virtual CPUs, attached block storage, and trusted computing platform), and then requests a VM launch. The Nova scheduler, in turn, selects the least-loaded physical host that meets the requirements and launches a VM with the selected image.

To demonstrate the role of cryptographic keys in an OpenStack solution, consider a social media website with a typical three-tier architecture, hosted on a public cloud. The tiers include a web front end for the user interface, an application middle tier for logic such as determining what data to display, and a database back end that holds authentication data and user-generated content such as images and comments. Neutron helps define a virtual private network for the collaborating VMs.

The multi-tenant public cloud in this example would host multiple customers, isolating them from each other. End users would expect secure connections to the site and protection against unauthorized access to their accounts. Likewise, the site's owners would expect protection against unauthorized access to their customers' personal information, as well as to the web pages themselves.

Key Management for OpenStack

Encryption of both end-user data and VMs facilitates protections such as those described above, helping ensure that data remains unreadable by intruders, even if the cloud itself were compromised. Controlling access to encryption keys and data using an authentication and authorization infrastructure can allow legitimate users to access the data, while restricting it from other cloud tenants and end users.

Generating strong encryption keys to protect objects and volumes requires a good source of random numbers. The keys themselves need to be encrypted for secure storage, and they must also be as readily accessible as the objects they encrypt, to avoid increasing object-access latency.

Intel, Mirantis, and Rackspace each worked independently on key management for OpenStack, with Intel posting a blueprint and use cases. At OpenStack Summit Portland in April 2013, consensus was built around the need to maintain key management as a separate service in its own git repository, incubating the project before finally including it in OpenStack. Specific design criteria established for the key manager included the following:

- **Provide a RESTful API** to create, save, retrieve, and destroy keys, with support for both symmetric and asymmetric keys, and keys of different length
- **Integrate with OpenStack** for authentication and access control
- **Provide audit logging** of key accesses and lifecycle events
- **Support high availability** for the key-management service

Development Status and Future Plans

Rackspace took leadership on developing the OpenStack key manager, initiating the Barbican open-source project. Intel provided integration with OpenStack's authentication and authorization system, made general design suggestions, and provided code-review support. The code repository and documentation are available at **https://github.com/cloudkeep/barbican** and **https://github.com/cloudkeep/barbican/wiki**, respectively.

Organizations that have joined the development effort include HP, the John Hopkins University Applied Physics Laboratory, and Red Hat. OpenStack will be using keys from Barbican for volume and block encryption, and Barbican is making steady progress toward becoming an official OpenStack project.

Looking ahead, the community intends for the key manager to add support for creation and certification of public-private key pairs, which could be certified either by an internal or external Certificate Authority (CA). Using public key infrastructure (PKI) to establish secure inter-service communications with SSL could simplify provisioning and cloud setup. The development of support for specifying and storing customer-default encryption preferences has not yet been undertaken.

Software support is planned for periodic background tasks such as handling expiring keys, refreshing master keys, replacing keys, and updating certificate status based on revocation lists. The key-management service will have a client component that can work against a plug-in Hardware Security Module (HSM).

The Key Management Interoperability Protocol (KMIP) will be examined in more detail to determine a minimal set that meets customer needs. The initial implementation currently supports formatter plug-ins to support various key formats, including KMIP.

Hardware Enablement from Intel

Software capabilities such as the OpenStack key manager work in conjunction with capabilities provided by Intel® architecture that enhance data protection in the cloud, while reducing or eliminating associated performance burdens. Features offered by current and recent generations of the Intel® Xeon® processor include the following:

- **Intel® Trusted Execution Technology (Intel® TXT)** enables solutions that help verify integrity of the key manager host's launch environment against rootkits and other compromises.
- **Intel® Digital Random Number Generator (DRNG)** is based in hardware, enhancing generation of strong keys in terms of speed, unpredictability, and resistance to software-based compromise.
- **Intel® AES New Instructions (Intel® AES-NI)** accelerate encryption in hardware, using the Advanced Encryption Standard (AES).

Conclusion

The key manager being developed within the Barbican project enables robust encryption to protect data in clouds based on the OpenStack environment. A vibrant development community and growing industry and academic support continue to add and refine capability. Intel is committed to making ongoing contributions that enhance the key manager as well as OpenStack more generally, while also innovating around hardware features, instruction sets, and programming techniques to enrich and accelerate encryption.

Now more than ever, OpenStack clouds eliminate compromises between data protection and performance.

Additional Resources

- **OpenStack home page:** www.openstack.org(http://www.openstack.org/)
- **Barbican open-source key-management project:** https://github.com/cloudkeep/barbican
- **Key Management Interoperability Protocol (KMIP) 1.1 specification:** https://www.oasis-open.org/committees/download.php/42878/kmip-spec-1.1-cd-01.pdf
- **Intel® Digital Random Number Generator (DRNG) software:** http://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide (http://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide)
- **Intel® AES New Instructions:** http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni/ (http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni/)

(http://facebook.com/sharer.php?u=https://01.org/blogs/tlcounts/2014/encryption-cloud-openstack-key-manager&t=Encryption for the Cloud with OpenStack* Key Manager)
(http://twitter.com/intent/tweet?url=https://01.org/blogs/tlcounts/2014/encryption-cloud-openstack-key-manager&text=Encryption for the Cloud with OpenStack* Key Manager)

- Privacy Policy
- Terms of Service
- Contact Us
- Jobs
- *Trademarks (http://www.intel.com/content/www/us/en/legal/trademarks.html)

- Cookies(http://www.intel.com/content/www/us/en/privacy/intel-cookie-notice.html)