Hermes PoC
by Cossack Labs

Cossack Labs release a proof of concept version of **Hermes** — a framework for cryptographically assured access control and data security.

A PoC reference implementation of **Hermes** is **Hermes-core 0.5.1**, the source code and accompanying documentation of which became available on December 13, 2017.

What is Hermes

Hermes is a cryptography-based method of providing protected data storage and sharing that allows enforcing cryptographically checked CRUD permissions to data blocks and doesn't let server that's running **Hermes** do anything worse than DoS.

Hermes operates with data that is subdivided into records, which represent the hierarchy of recordsets and groups of recordsets. Each blob of data is encrypted using a symmetric key, which a set of hashes is generated from. Possession of a symmetric key by a user allows reading and carrying out other processes on hashes (including with writing the data).

Hermes allows distributing only the necessary amount of data between the system components and, consequently, limits the possible damage that may be done if a component is compromised.

Hermes enables collaboration and distributed data sharing through enforcing access control with the help of cryptographic methods (both public key cryptography and traditional symmetric cryptography). **Themis** by Cossack Labs is used as a cryptographic services library, providing well-known and widely used primitives to implement cryptography.

Advantages

As the network design of **Hermes** can be mapped to a typical client-server architecture, the following major advantages of **Hermes** can be defined:

1) An absence of a central point of failure (sensitive data compromisation);
2) Restriction of access to cryptographic keys and sensitive data in plain text for the server side;
3) End-to-end authenticated encryption between all the components (both server side and client side).

Components of Hermes

There are 3 storage entities in **Hermes** (and, consequently, in **Hermes-core 0.5.1**) that constitute the Server side:

- Data store that contains the hierarchy of encrypted objects.
- Credential store that stores keys and hashes, asymmetrically encrypted in such a way that can only be decrypted by authorised user's private key.
- Keystore that contains the symmetric keys (for READ and UPDATE), with as many copies of these keys as there are users authorised to access the record, where every copy is wrapped (asymmetrically encrypted) with a public credential of the respective authorised user.

The 4th entity of **Hermes** is the Client. Client (or clients) is the active entity in the Hermes architecture, the one that actually produces or consumes the data. The Client only possesses the keypair that allows decrypting the asymmetrically encrypted data from the Server.

Availability

Hermes-core, the proof of concept implementation, is fully available on C, however, reference client implementations are available in C, Python and Go:
Hermes-core has high-level wrappers to make Hermes API available on:

- C;
- GoLang v.1.9;
- Python 2.7.12, 3.3.6, 3.4.4, 3.5.3, 3.6.2.

We have verified that Hermes-core package builds on:

- Debian "Wheezy" (Debian 7)
- Debian "Jessie" (Debian 8)
- Debian "Stretch" (Debian 9)
- Ubuntu Trusty Tahr (Ubuntu 14.04)
- Ubuntu Xenial Xerus (Ubuntu 16.04)
- Ubuntu Yakkety Yak (Ubuntu 16.10)
- Ubuntu Zesty Zapus (Ubuntu 17.04)
- CentOS: 7

Hermes-core library can be installed from our repository or built from sources.

Typical use-cases

The typical cases for which Hermes would be the most beneficial:

Cryptographic access control
Hermes can be deployed for building cryptographic access control in your application. With its help you can regulate the READ and WRITE access rights through a cryptographic scheme resistant to privilege escalation.

Secure collaboration and data sharing
Hermes is a cryptography-based method of providing protected data storage and sharing that allows the enforcement of cryptographically-checked permissions between any number of Hermes clients.

Multi-user object store
Using Hermes, you can build end-to-end secure document/object stores where every document or field's access rights can be granted to any registered user of the system, transparently, and with low overhead.

Additional information on Hermes

Please see "Hermes – a framework for cryptographically assured access control and data security" scientific paper for a more detailed theoretical explanation of concepts behind Hermes and "Implementing Hermes-based Security Systems" document for additional information on implementing and changing Hermes-core.

About Cossack Labs

Cossack Labs Limited is a privately funded, British company headquartered in London, England. Our R&D team is located in Kyiv, Ukraine.

Cossack Labs create data security tools and services for software developers. We provide sophisticated cryptographic defences for modern systems, without sacrificing security or usability. Our core technologies are available as open source software and we provide separate commercial licensing and support services.

Get Hermes-core now

Get in touch