Functional Encryption Technologies

Yolan Romailler
@AnomalRoil

November 2019 — ISSE

Starting with encryption

Public-key encryption is ubiquitous on internet!

Be it HTTPS, SSH, messaging, voip, storage, … everything seems to rely on public-key schemes to ensure security nowadays.

But encryption is usually seen as being an **all or nothing** option:

- either you have it,
- or you don't!

In a nutshell, functional encryption is all about challenging that idea.

Functional Encryption (FE)

- Functional encryption was first proposed by Amit Sahai and Brent Waters in 2005.

- Then Dan Boneh, Amit Sahai and Brent Waters formalized the notion of functional encryption in a paper in 2010.

- We can say FE is a **public-key encryption scheme** with different decryption key**s** allowing to learn the result of a specific function of the encrypted data.

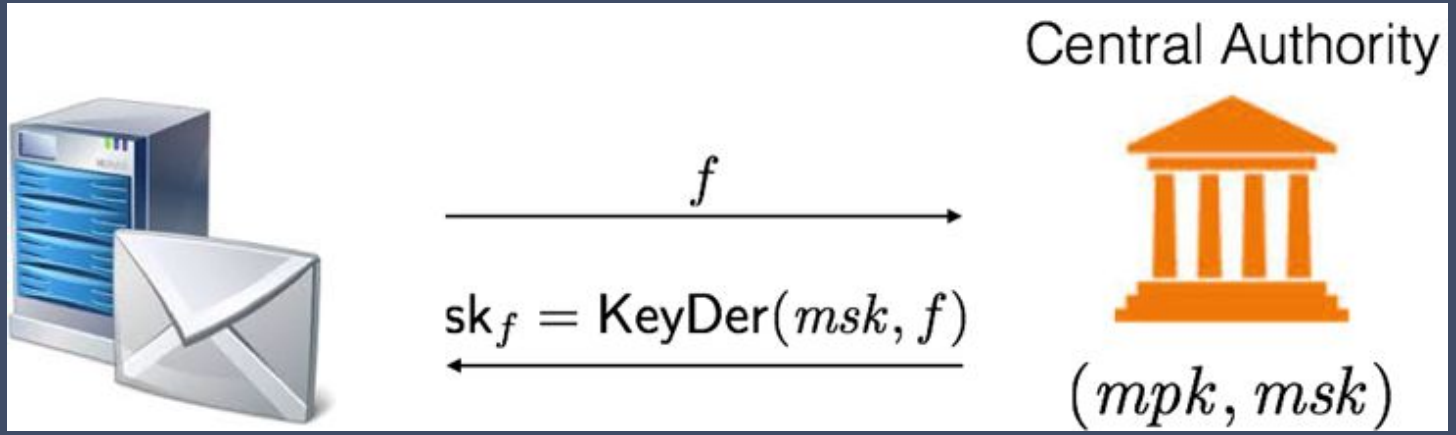
What is FE all about?

Before FE, it was accepted that:

- Public-key encryption is a method to send secret data **to a single entity** holding a given secret key corresponding to the public key.
- Access to the encrypted data is **all or nothing** – one can either decrypt and read the entire plaintext or one learns nothing about the plaintext.

Functional encryption tries to change these!

More details about the setup of FE schemes

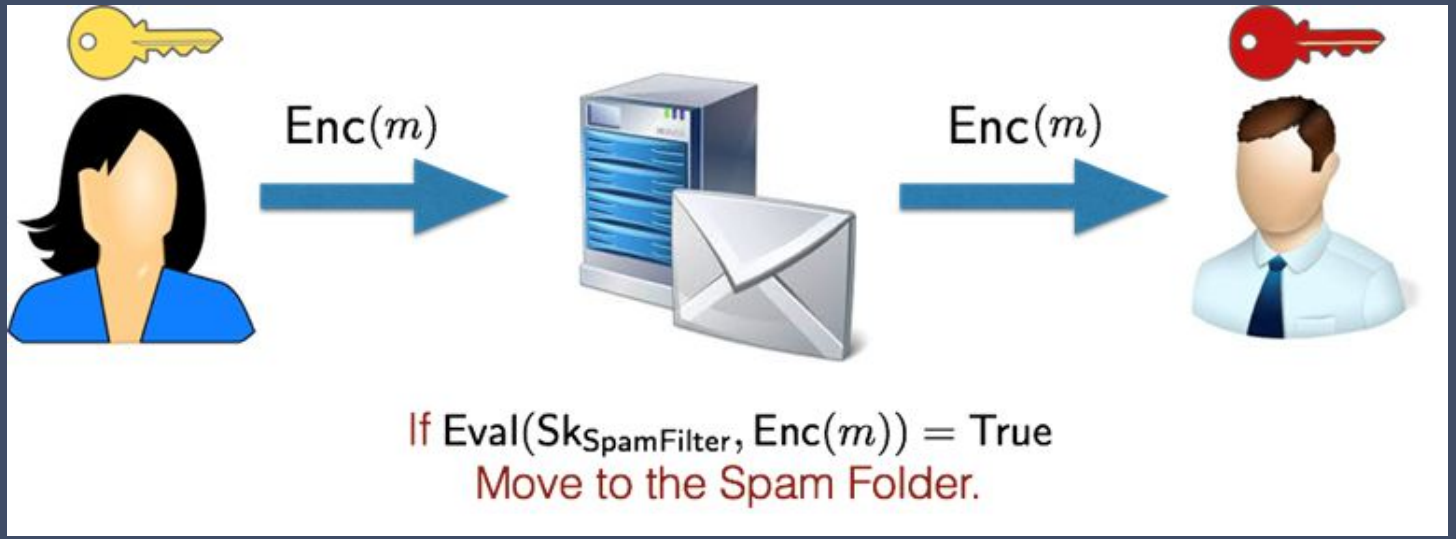


Why FE?

There are many different scenario in which we might want to evaluate a specific function on encrypted data without needing access to the plaintext data!

- Processing of data by a partially trusted 3rd party
- Delegating complex computation to remote, powerful servers
- Having a fine-grained access control on the encrypted data (who has access to what, and what they can compute)

Classical example



A couple examples of use-cases

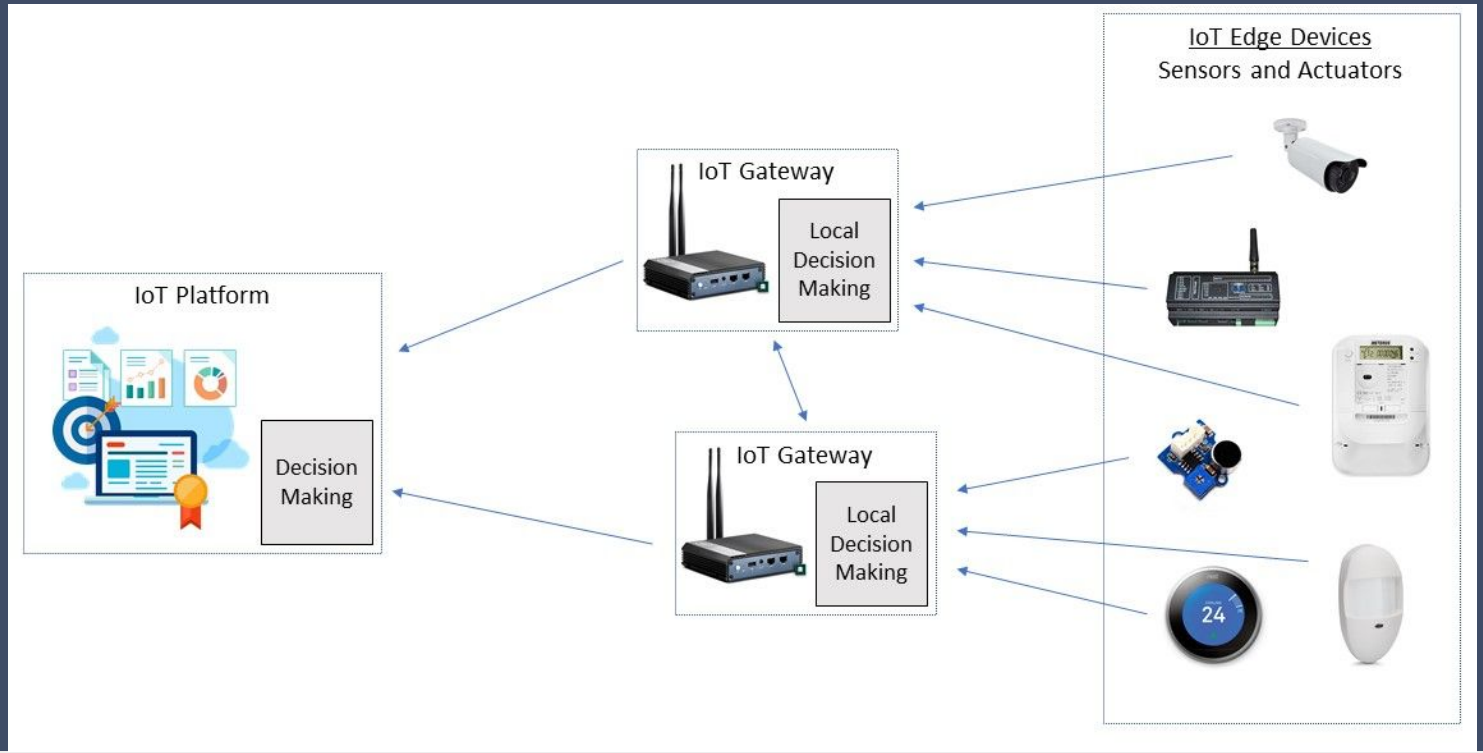
FE allows to **take decisions at the gateway level**, even if the data in transit is **end-to-end encrypted** for the backend system.

One example: **local decision making in a smart grid network**.

Compliance requires end-to-end encryption between the smart meter in your house and the backend systems of the electricity provider.

You might want to take decision earlier at the gateway level if something goes wrong on the network → FE allows to do so.

Local what?


IoT Edge Devices
Sensors and Actuators
IoT Gateway
Local Decision Making
IoT Platform
Decision Making
IoT Gateway
Local Decision Making
24

Another example

Machine learning on encrypted data!

Not even kidding:

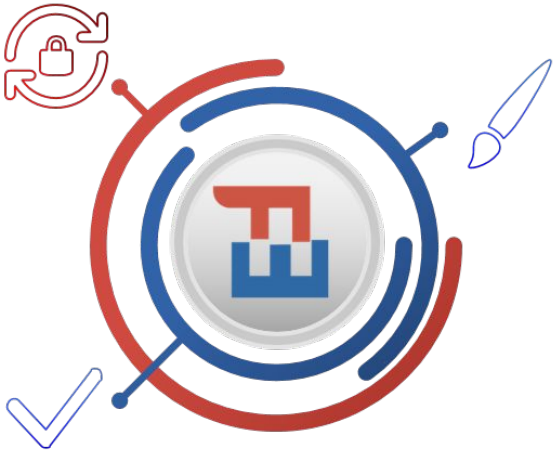
Try it out:

https://github.com/fentec-project/neural-network-on-encrypted-data

which is implementing https://eprint.iacr.org/2018/206

The goals of the FENTEC project

Implement a unified **cryptographic API** of Functional Encryption systems



Design **functional encryption systems** with varying functional, security, **hardware** and **software** requirements

Validate and **demonstrate** FENTEC technologies and **solutions**

FENTEC's consortium

Future works

- Hardware accelerators
- Richer functionalities
- Quantum-safe schemes
- Integrating more FE schemes into the FENTEC library
- Implementing prototypes that use of the FENTEC library
- Getting you to use FE technologies! ;)

Follow me on Twitter: @AnomalRoil

Some links!

- http://fentec.eu/
- https://github.com/fentec-project/gofe
- https://github.com/fentec-project/CiFEr
- https://github.com/fentec-project/neural-network-on-encrypted-data

- https://research.kudelskisecurity.com

Food for thoughts & discussions

- The need for strong encryption & privacy does not rule out entirely the need for regulation & auditability, as such FE can help achieve both.
- When designing new cryptographic capabilities, should we be worried about their potential for “dual use” (i.e. both civilian and military applications)? If so, how are we to go about it?
- Additive notation is better for discrete logarithm protocols ;)
- FE is not a good primitive for your next blockchain projet.