rnp

Confium: an open-source framework to support threshold cryptography standardization

NIST MPTS 2020

November 5, 2020

Ronald Tse, Ribose

Jointly prepared by Daniel Wyatt, Nickolay Olshevsky, Jeffrey Lau

riboseopen

Mozilla Thunderbird’s OpenPGP email is powered by RNP

SEP 7 2020

OpenPGP in Thunderbird 78

Ryan Sipes

Updating to Thunderbird 78 from 68

Soon the Thunderbird automatic update system will start to deliver the new Thunderbird 78 to current users of the previous release, Thunderbird 68. This blog post is intended to share with you details about our OpenPGP support in Thunderbird 78, and some details Enigmail add-on users should consider when updating. If you are interested in reading more about the other features in the Thunderbird 78 release, please see our previous blog post.

Updating to Thunderbird 78 is highly recommended to ensure you will receive security fixes, because no more fixes will be provided for Thunderbird 68 after September 2020.

The traditional Enigmail Add-on cannot be used with version 78, because of changes to the underlying Mozilla platform Thunderbird is built upon. Fortunately, it is no longer needed with Thunderbird version 78.2.1 because it enables a new built-in OpenPGP feature.

- RNP is a high-performance OpenPGP library
- Thunderbird 78+ embeds RNP for its end-to-end email encryption functionality
- Only major email client with native OpenPGP functionality
- 35M+ Thunderbird installations
- Open, freely licensed (BSD)
- Audited by Cure53, monitored by Google’s OSS-Fuzz program

riboseopen

Plenty of steps required to apply TC in Thunderbird

- Vast gaps to overcome between cryptographic research and practical deployment
  1. Research
  2. Standardization at SDOs
  3. Adoption by cryptographic libraries (and implementations)
  4. Developer education and end-user application adoption

- Cross-platform: abstracted primitives and resources
  - Computation, algorithms, smartcards, HSMs, networking

- => Decouple cryptographic design, implementation, distribution and adoption

Confium is a cross-platform trust store that bridges cryptographers with practical cryptography usage

- Generalized environment with abstracted primitives for cryptographers to develop prototypes to production algorithms and schemes

- *Supports the standardization efforts of threshold cryptography at NIST*
  - Common API for TC implementations/primitives
  - Enables real-world end-user application testing
  - Simplify comparisons by providing a level-playing field
  - Supplied primitives (e.g. networking) lets cryptographers focus on what's important

- Open source, openly licensed!

riboseopen

Confium provides an abstraction that supports new cryptographic families

- Provide a *generalized platform for cryptographic implementations*
  - cryptographic execution environment
  - primitives for networking, other OS resources
  - distributed and remote resources
- Extensible architecture for new cryptographic usage
  - crypto provider plugins to bridge existing crypto libraries
  - crypto storage plugins to support different keys types and parameters
- Platform-independent, compartmentalized key storage
- Gives applications control of extension activity

RNP/Confium receives funding support by MOSS and NLNet

- Mozilla Open Source Support
  - Foundational Technology award
  - Secure Open Source award

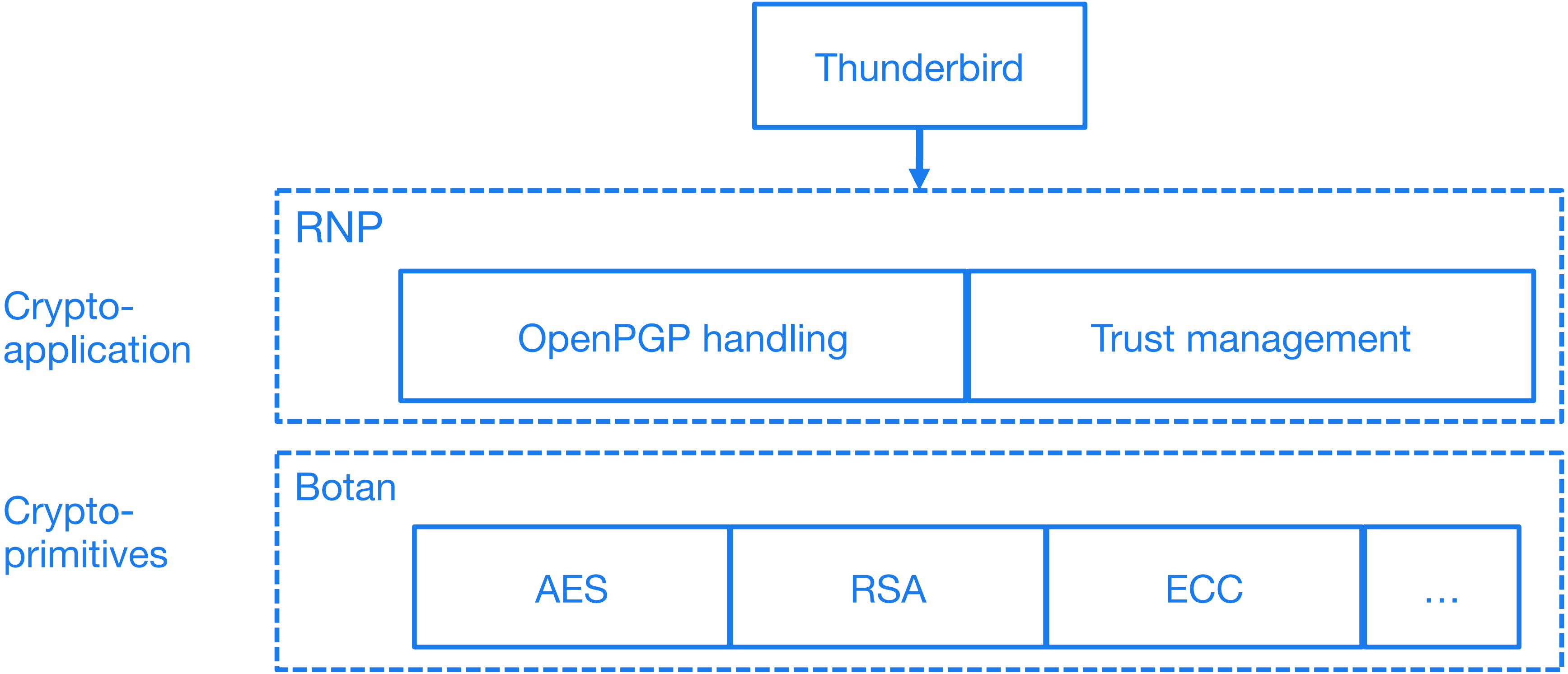
- NLNet Next Generation Internet
  - NGI Zero Privacy Enhancing Technology award
  - Supported by funding from EU's Horizon 2020 programme under grant agreements No 825310 and 825322

moz://a

NGI ZERO

European Commission

Current user application architecture (without Confium)



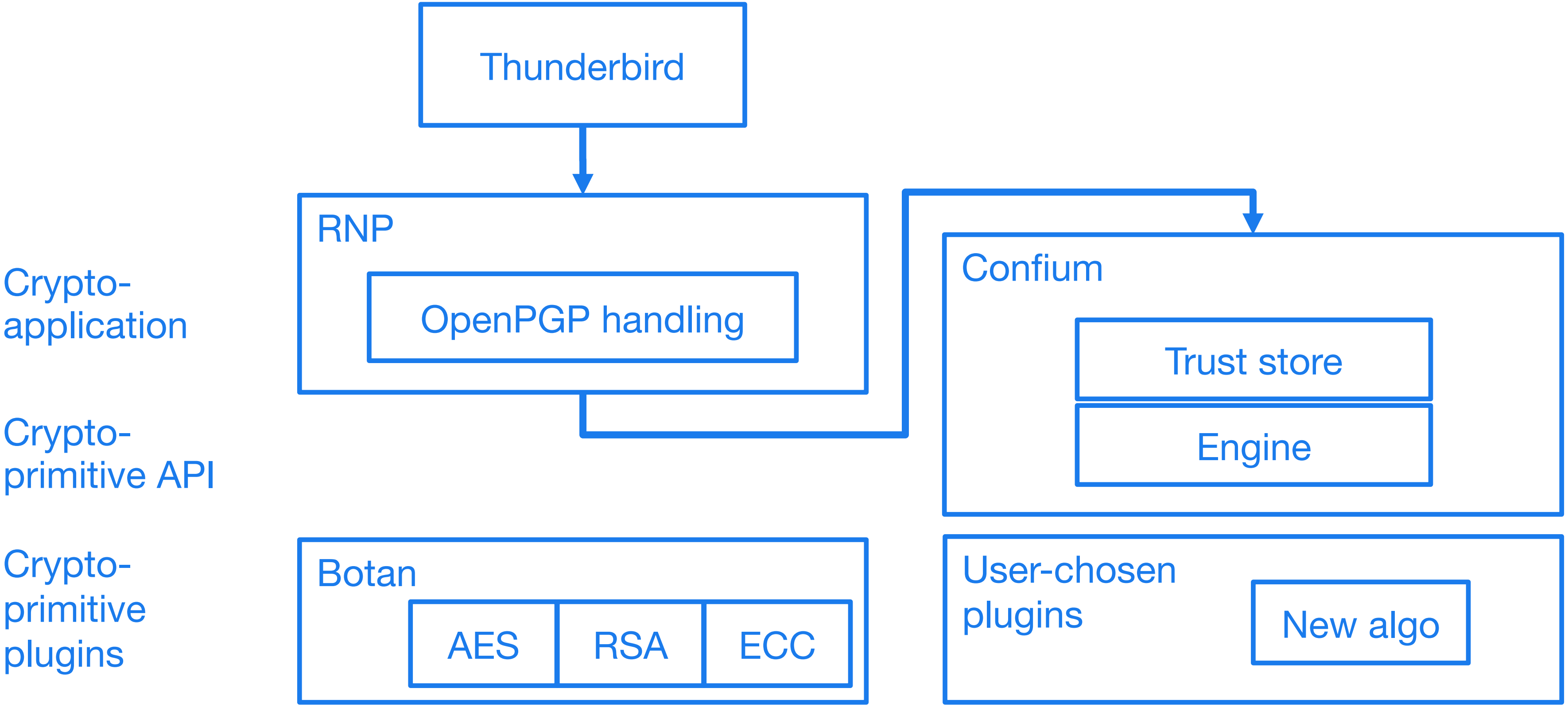
riboseopen

Confium unhinges user applications from cryptography implementations

- Support smart cards and other trust mediums
- Retrieve keys from external sources
- No longer bound to particular algorithm implementations from underlying cryptographic libraries
  - e.g. addition of plaintext padding in an updated version can screw the stack
- Extendable to future cryptographic families

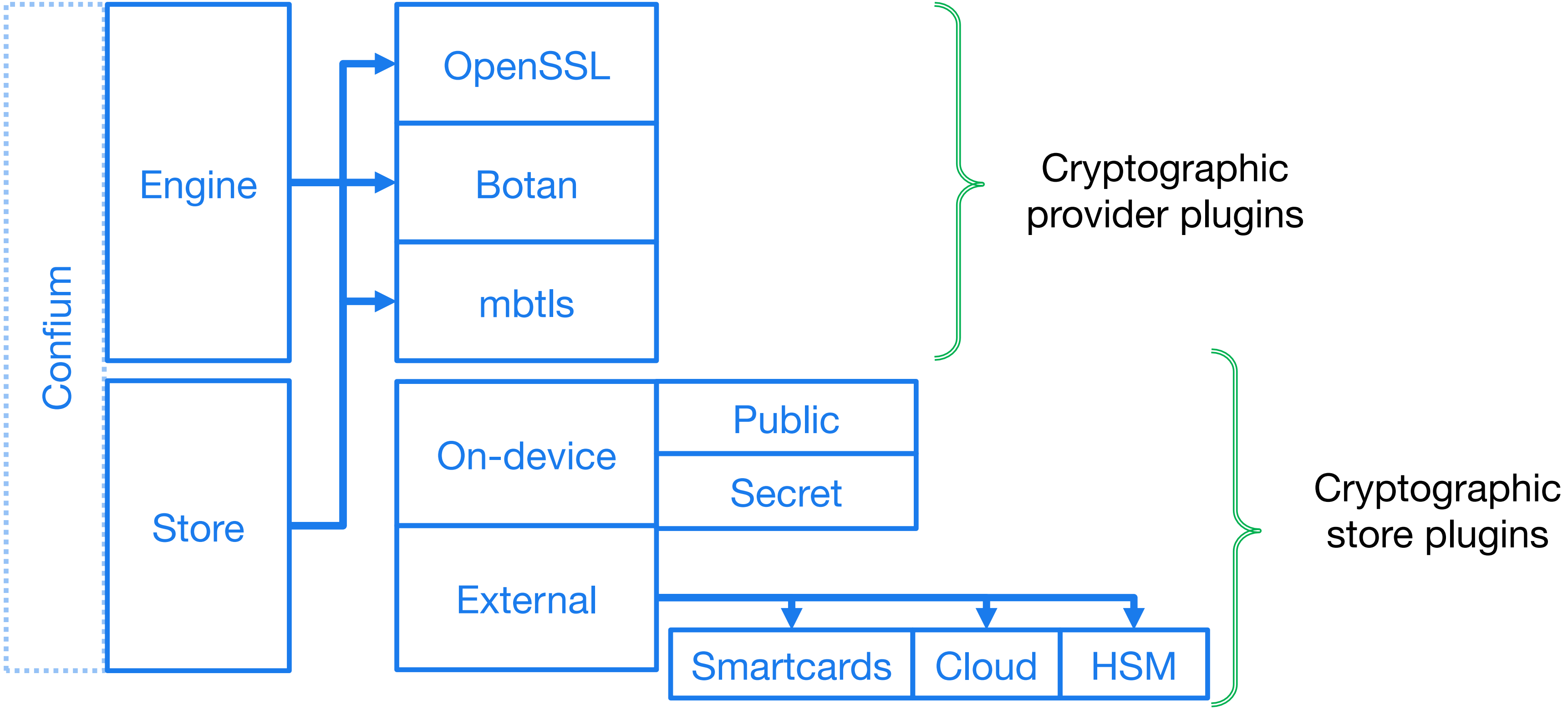
- => Enables cryptographers to directly contribute to the trust store

Re-architected user applications on Confium



riboseopen

Integrated architecture providing a unified trust/crypto layer



riboseopen

Cryptographic plugins ensures future extensibility as an isolation layer across cryptographic libraries

- Provider plugins
  - Type 1: purely implemented in Confium via FFI
  - Type 2: hybrid implementation via FFI in Confium, that utilizes existing implementations from cryptographic libraries (e.g. OpenSSL, mbtls)
  - Type 3: wrapper implementations of existing implementations from cryptographic libraries
- Store plugins
  - Different implementations for new secret/key types

- Publishers manage their own plugins
- Users decide what to install

Example: prototyping and production of crypto-primitives

- Each plugin needs some way of specifying dependencies

```
require "confium/ffi"
require "confium/openssl/1.1.1"

class ClownRsa::Key
  def generate
    process OpenSSL::RSA.generate_key
  end

  def sign(data)
    ...
```

```
include <confium/ffi.h>
include <confium/openssl/1.1.1.h>

namespace ClownRSA {
namespace Key {

RSA generate() {
  const int kBits = 1024;
  const int kExp = 3;
  return OpenSSL::RSA_generate_key(
    kBits, kExp, 0, 0);
```

Call for interest

- Reiteration of goals:
  - Assist upcoming NIST standardization efforts
  - Bring your algorithm to real-world user applications

- Example: https://github.com/rnpgp/confium/blob/wip/example.rb
- Seeking interest from cryptographers to test drive Confium!
- Contact the Confium team at confium@ribose.com

Thank you, questions welcome!

riboseopen