A New Kid on the Block: CLINT - a Cryptographic Library for the INternet of Things

Mike Scott

CertiVox Ltd

Real World Crypto, London, January 2014

A problem

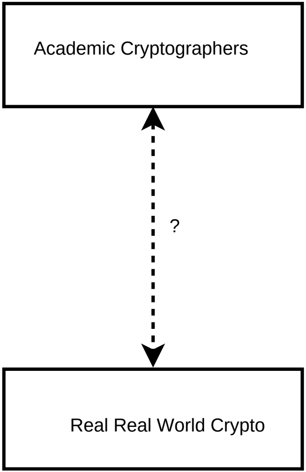

Figure: Communication Problem

Part of the Reason?

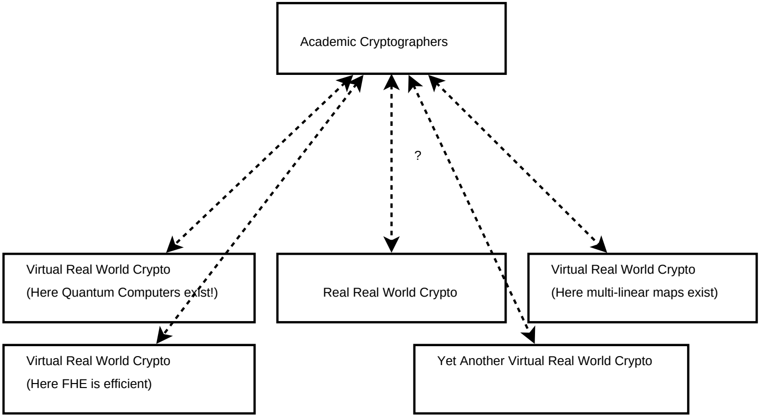

Figure: Research Reality

There are Real Problems!

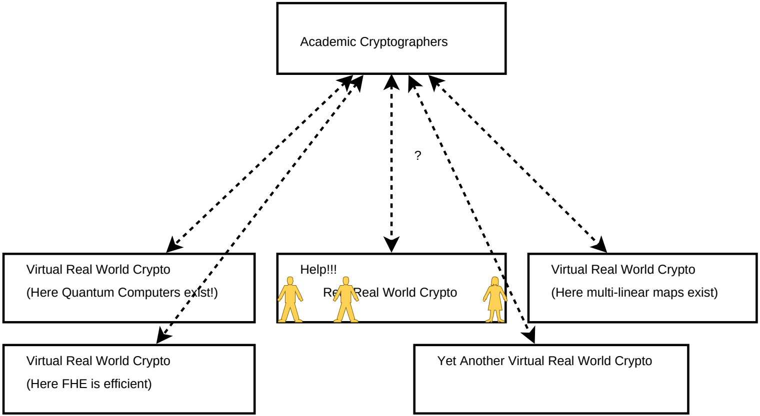

Figure: These guys need help!

Maybe Part of the Solution

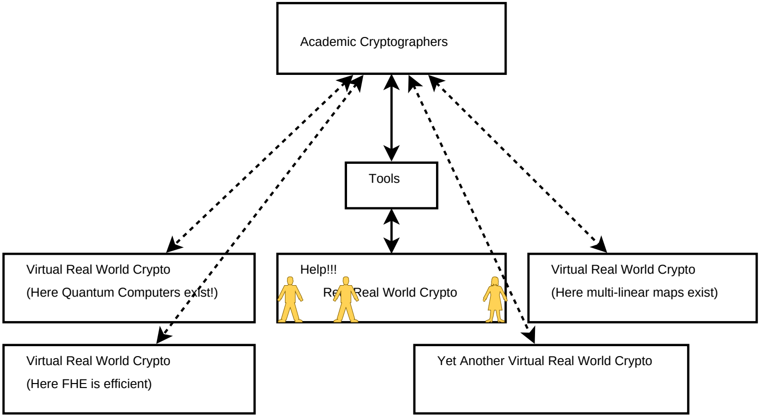

Figure: Easy to use tools

Existing Crypto Libraries

- There are many crypto libraries out there.
- Many offer a bewildering variety of cryptographic primitives, at different levels of security.
- Many use extensive assembly language in order to be as fast as possible.
- Many are very big, even bloated. Some rely on other external libraries.
- Most were designed by academics for academics, and so are not really suitable for commercial use.

CLINT – 1

- CLINT is completely self-contained (except for the requirement for an external entropy source).
- CLINT is for use in the pre-quantum era – that is in the here and now.
- CLINT is portable - no assembly language.
- The release version is available in pure C, Java and Javascript using only generic programming constructs.
- New language version can be produced in 3-4 weeks. Next up C# and Swift.
- All versions will be “identical” – all internal calculations are the same.

CLINT – 2

- CLINT is fast, but does not attempt to set speed records (a particular academic obsession).
- CLINT is small – less than 10,000 lines of code.
- CLINT has a very small footprint – important for IoT.
- CLINT supports only one level of security (AES-128)
- CLINT implements only curve based Public Key methods (including Pairings)

CLINT – 3

- Support for SHA256, AES-128, AES modes plus GCM
- Raw Entropy processing for random number generation.
- Elliptic Curves (Weierstrass, Edwards, Montgomery)
- Types of moduli (general, Montgomery friendly, pseudo-mersenne)
- BN-curve based optimal pairings
- 2048-bit RSA (legacy support)

CLINT – 4

- Awareness of modern pipelined architecture
- Avoid **if** statements (particularly unpredictable branches)
- Side channel attack resistance baked-in.
- Example APIs that communicate to the "Real World" using simple byte arrays.

Raspberry pi implementation - space

	Code Size	Maximum Stack Usage
ECC protocol -O3	63236	3004
ECC protocol -Os	30102	2940
PBC protocol -O3	80493	10124
PBC protocol -Os	45008	9744

Table: Typical Memory Footprint

Raspberry pi implementation - time

	Time in milliseconds
ECC point multiplication -O3	11.9
ECC point multiplication -Os	17.2
PBC pairing -O3	85
PBC pairing -Os	122

Table: C Benchmarks

Question Time

- Thank you for your attention