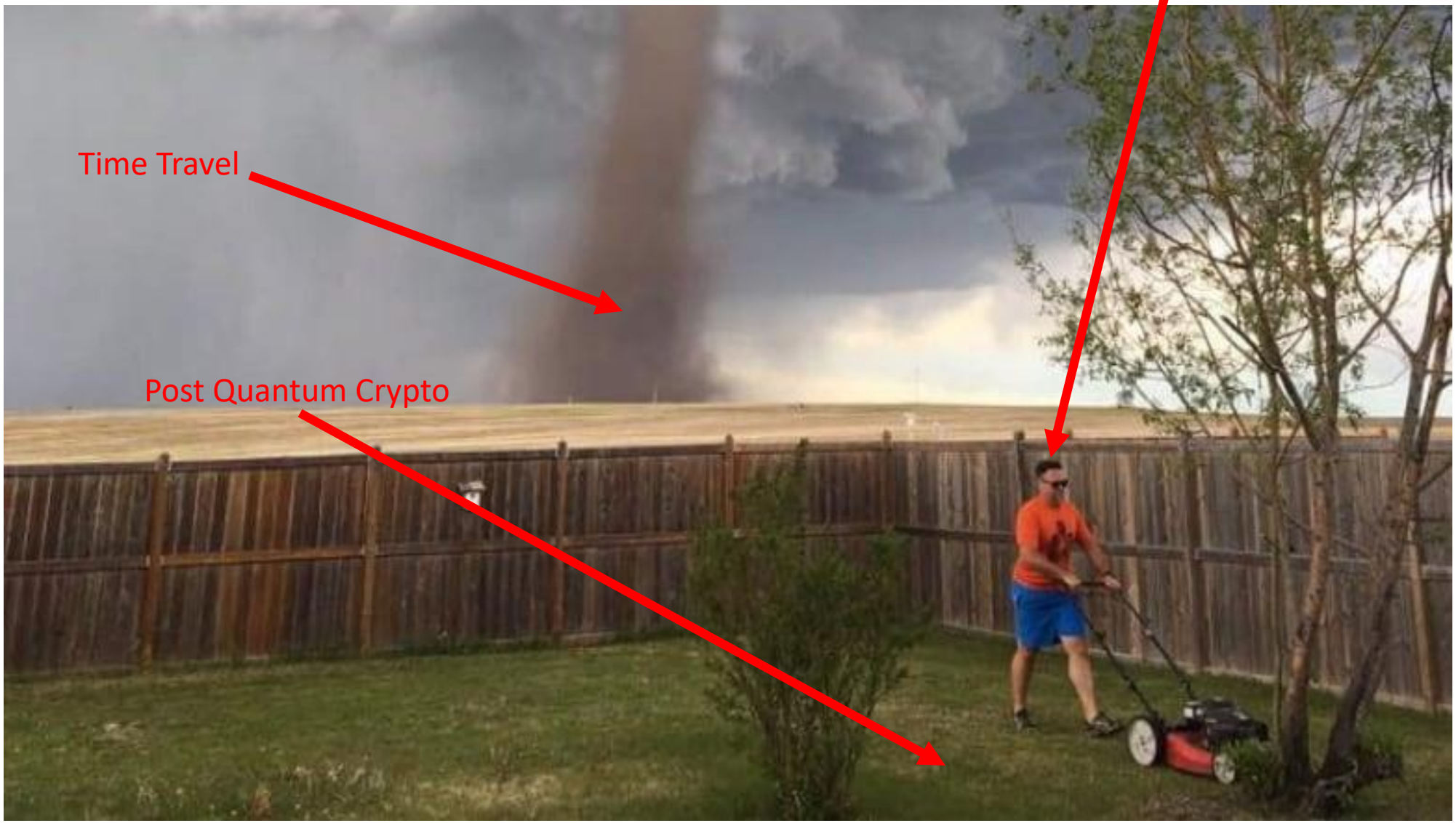
A Call for Time Travel Resistant Crypto (TTRC)

Colin O’Flynn

The World will Judge Us.



Why does PQC get all the love?

- Post Quantum Crypto – the existence of sufficiently powerful quantum computers is an open problem.

- But *if* those computers (devices) exist – very bad time in store.

- That *sounds a whole lot* like something else.

Where are the time travelers?

- Common statement – if time travel exists, wouldn't we see them?
  - Maybe we don't know of them.
  - Maybe our time isn't too interesting to visit.
  - Maybe time travel can *only transfer small particles or just data*.

Immediate Applications of TTRC

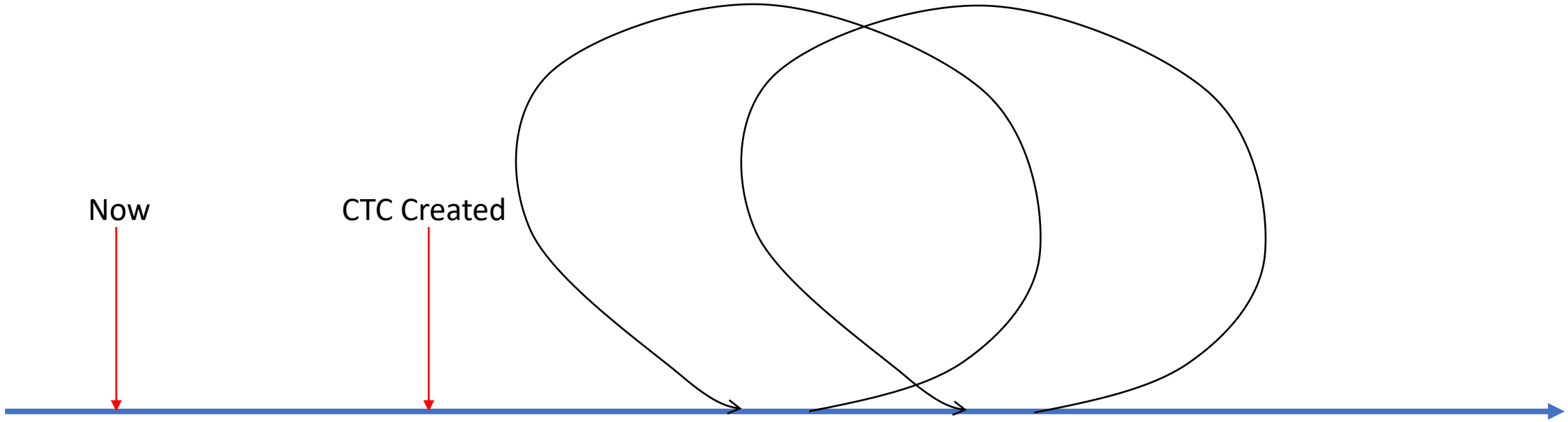
- Time AI (.io) – Uses Quantum Cryptography that entangles a random key stream from the *past* and *future* [1]
  - Published by Cornell University*
  - Presented at Black Hat USA 2019+

* Published on arXiv.org

+ Sponsor session at cost > $50k USD

[1] http://timeai.io

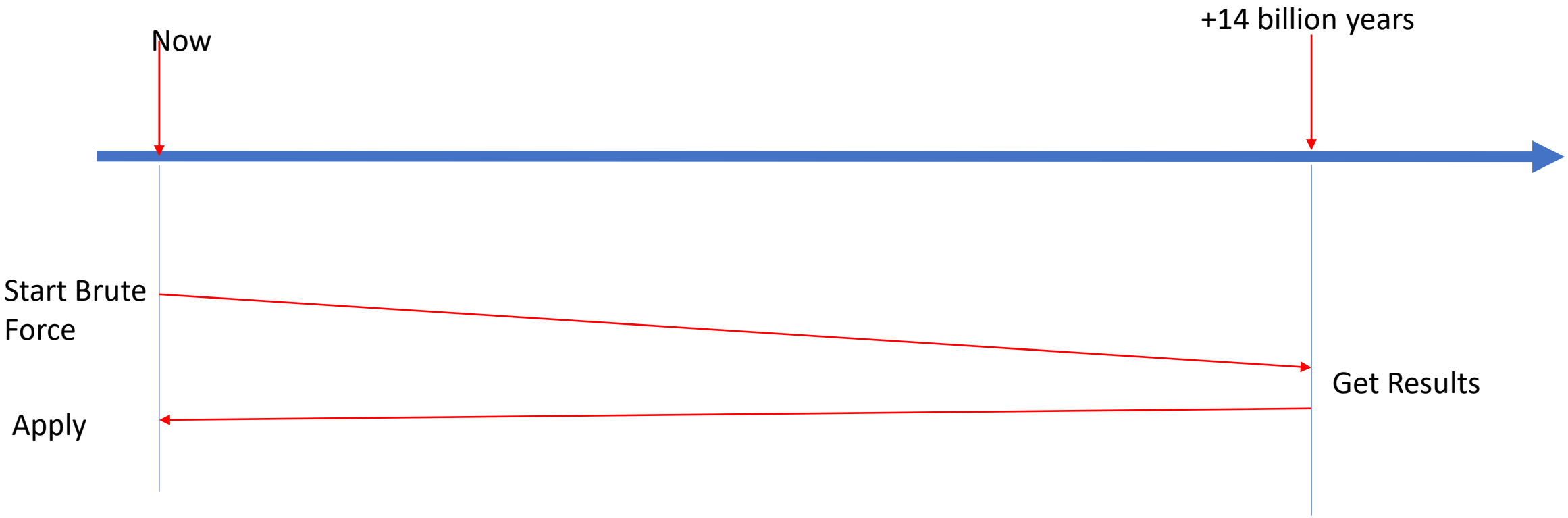
Does time-travel (TT) require a Closed timelike curve (CTC)[2]?



Time travel only possible once CTC is created (or new timeline?)

Attacks that TTRC must survive.

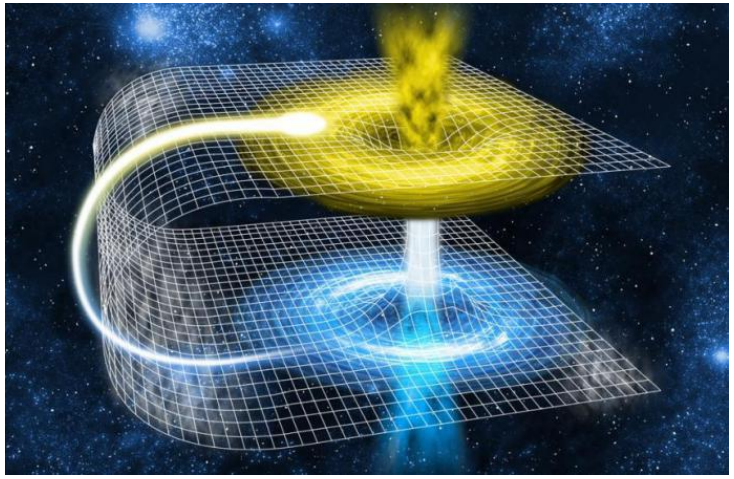
- Brute-force with TT assistance



Attacks that TTRC must survive.

- Transfer of key material to later time.

- Assume we can target the value of some physical bits…
- If we know where HSM (was) located, could we recover key material out of bus?



Other attacks TT makes possible

- Evil Cryptographer Attack
  - Could someone *we know be* a time-traveler? What would that person look like?

Other attacks TT makes possible

- Evil Cryptographer Attack
  - Could someone *we know be* a time-traveler? What would that person look like?

- Contributes work in a *suspiciously open and free manner*.
- Widely used in industry, with vague understanding by said industry of the actual algorithms.

NaCl: Networking and Cryptography library

Computer Aided Cryptography Engineering

ECRYPT II

Introduction

NaCl (pronounced "salt") is a new easy-to-use high-speed software library for network communication, encryption, decryption, signatures, etc. NaCl's goal is to provide all of the core operations needed to build higher-level cryptographic tools.

Of course, other libraries already exist for these core operations. NaCl advances the state of the art by improving security, by improving usability, and by improving speed.

The following report contrasts NaCl with other libraries from a security perspective: (PDF) Daniel J. Bernstein, Tanja Lange, Peter Schwabe, "The security impact of a new cryptographic library". Pages 159–176 in Proceedings of LatinCrypt 2012, edited by Alej Hevia and Gregory Neven, Lecture Notes in Computer Science 7533, Springer, 2012. ISBN 978-3-642-33480-1.

The following report was created for Research Plaza and gives an introduction to NaCl for a wider audience: (PDF)

Upcoming features

Major features in the next release of NaCl: full PIC support, for easy integration into other languages; Ed25519 signatures (currently available in SUPERCOP); NEON optimizations.

Contributors

The core NaCl development team consists of Daniel J. Bernstein (University of Illinois at Chicago and Technische Universiteit Eindhoven), Tanja Lange (Technische Universiteit Eindhoven), and Peter Schwabe (Radboud Universiteit Nijmegen).

NaCl was initiated under the CACE (Computer Aided Cryptography Engineering) project funded by the European Commission's Seventh Framework Programme (FP7), contract number ICT-2008-216499, running from 2008 through 2010. CACE activities were organized into several Work Packages (WPs); NaCl was the main task of WP2, "Accelerating Secure Networking". Work on NaCl at Technische Universiteit Eindhoven between 2008 and 2010 was sponsored by CACE.

to Cryptographic Operations and Primitives) API developed for the eBACS (EC
phy project funded by the U.S. National Science Foundation, grant number 07164
was sponsored by these grants. "Any opinions, findings, and conclusions or

Curve25519 is a state-of-the-art Diffie-Hellman function suitable for a wide variety of applications.

Given a user's 32-byte secret key, Curve25519 computes the user's 32-byte public key. Given the user's 32-byte secret key and another user's 32-byte public key, Curve25519 c

How do I use Curve25519 in my own software?

My `curve25519` library computes the Curve25519 function at very high speed. The library is in the public domain. You can and should include it in your own programs, rather t

Getting started. Download and unpack the `curve25519` library:

```
wget http://cr.yp.to/ecdh/curve25519-20050915.tar.gz
gunzip < curve25519-20050915.tar.gz | tar -xf -
```

To get an idea of how the library is structured, compile it:

```
cd curve25519-20050915
env CC='gcc -O2' make
```

Make sure to use appropriate compiler options for your platform, such as `-m64` for the UltraSPARC. The library will refuse to compile if it doesn't pass some stringent internal t
chips, such as the Pentium and Athlon, and it isn't fully optimized for those chips. But it does hold a bunch of speed records already.)

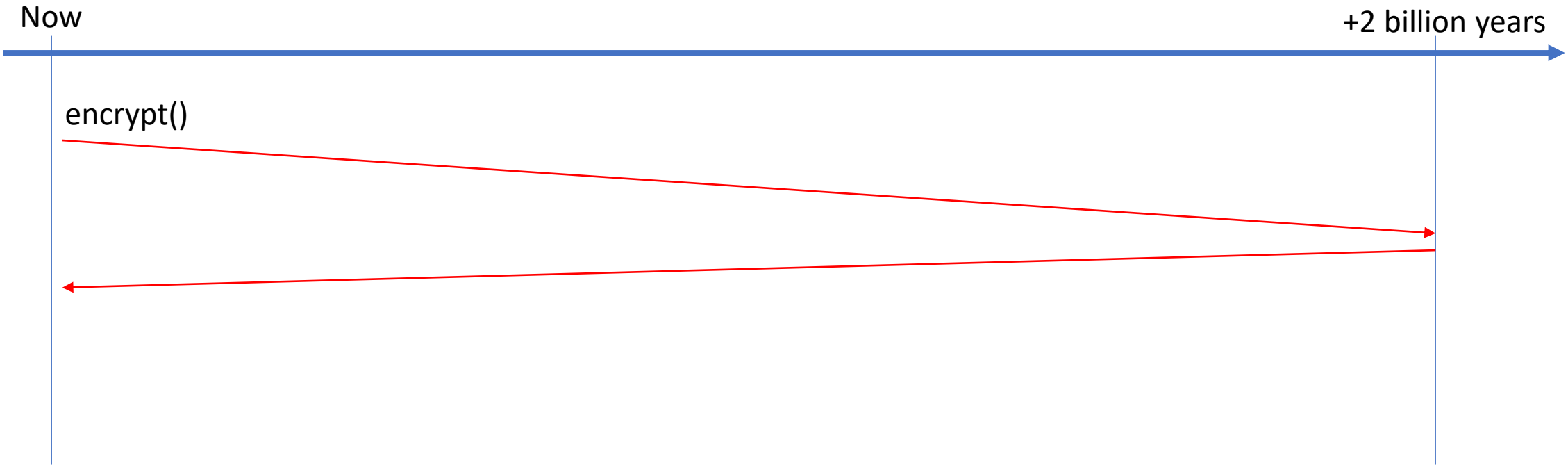
Copy the library source files into your project:

```
cp `cat FILES.lib` yourproject/
cat Makefile.lib >> yourproject/Makefile
```

For any C program that will use Curve25519, modify the program to include `curve25519.h`; also modify your `Makefile` to link the program with `curve25519.a` and to declare th

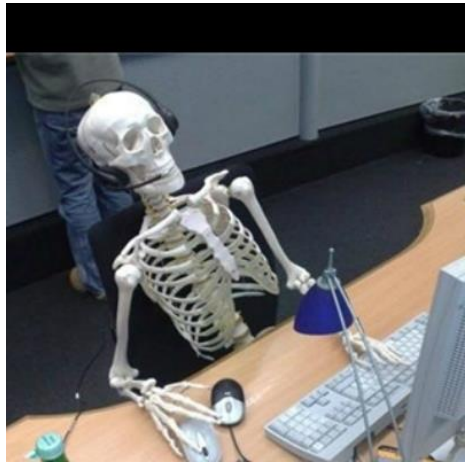
Simple solution for TTRC

- Define *now* our API… assume future solves problem (need to store bits somewhere future can access them).
- Need to incentivize future selves (is any data now valuable in the future?).



Requires a hardware solution (→ CHES 2021 topic?)

TTRC for Future Proof



- Post-quantum crypto is not TTRC.
- Fundamentally TTRC is future-proof.
- New architectures needed (non-causal cryptography?).

Find answers to your questions on TT: https://plato.stanford.edu/entries/time-travel/

[2] Gödel, Kurt, 1949 [1990a], "An example of a new type of cosmological solutions of Einstein's field equations of gravitation", in *Kurt Gödel: Collected Works* (Volume II), Solomon Feferman, et al. (eds.), New York: Oxford University Press, 190–8; originally published in *Reviews of Modern Physics*, 21 (1949): 447–450.