Algebraic Replicated Data Types: Programming Secure Local-First Software (Artifact)

Christian Kuessner
Technische Universität Darmstadt, Germany

Ragnar Mogk
Technische Universität Darmstadt, Germany

Anna-Katharina Wickert
Technische Universität Darmstadt, Germany

Mira Mezini
hessian.AI, Darmstadt, Germany
Technische Universität Darmstadt, Germany

Abstract

This work is about programming support for local-first applications that manage private data locally, but still synchronize data between multiple devices. Typical use cases are synchronizing settings and data, and collaboration between multiple users. Such applications must preserve the privacy and integrity of the user's data without impeding or interrupting the user's normal workflow – even when the device is offline or has a flaky network connection.

From the programming perspective, availability along with privacy and security concerns pose significant challenges, for which developers have to learn and use specialized solutions such as *conflict-free replicated data types* (CRDTs) or APIs for centralized data stores. This work relieves developers from this complexity by enabling the direct and automatic use of algebraic data types – which developers already use to express the business logic of the application – for synchronization and collaboration. Moreover, we use this approach to provide end-to-end encryption and authentication between multiple replicas (using a shared secret) that is suitable for a coordination-free setting.

This artifact demonstrates the approach in the context of a realistic case study. It shows that an implementation of the approach can handle realistic workloads, that the size of the data types does not grow indefinitely, and that it is feasible to always enable encryption for the intended scenario.

2012 ACM Subject Classification Information systems → Data management systems; Computer systems organization → Dependable and fault-tolerant systems and networks; Security and privacy → Cryptography

Keywords and phrases local-first, data privacy, coordination freedom, CRDTs, AEAD

Digital Object Identifier 10.4230/DARTS.9.2.26

Related Article Christian Kuessner, Ragnar Mogk, Anna-Katharina Wickert, and Mira Mezini, "Algebraic Replicated Data Types: Programming Secure Local-First Software", in 37th European Conference on Object-Oriented Programming (ECOOP 2023), LIPIcs, Vol. 263, pp. 14:1–14:33, 2023.
`https://doi.org/10.4230/LIPIcs.ECOOP.2023.14`

Related Conference 37th European Conference on Object-Oriented Programming (ECOOP 2023), July 17–21, 2023, Seattle, Washington, United States

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2023 Call for Artifacts and the ACM Artifact Review and Badging Policy.

1 Scope

The overall artifact is a OVA VM image containing:

- the benchmarks (runnable) with graph generation pipeline (as a Jupyter notebook)
- the source code for the benchmarks (including our library of CRDTs) with the compilation pipeline to produce the runnable benchmark code
- the runnable UI example application (including source code and compilation pipeline)

We claim that the overall artifact is functional and available, and that the source library and compilation pipeline is reusable.

2 Content

The VM should directly boot into a desktop environment with all relevant data on the desktop (folder: `/home/ecoop/Desktop`). The VM user is `ecoop` with password `ecoop`.

3 Functional

The benchmark figures can be recreated by:

- Double click the `RUN BENCHMARKS.sh` and select "execute in terminal".
- this should take about 10 15 minutes (on a modern laptop, or 5 year old desktop)
- note: this first runs JMH based performance evaluation (which includes a estimated time), then we run evaluation on the sizes in parallel: you will see intermixed output, but measurements generally go up to 10k elements, after which the benchmark finishes
- results are put into the `benchmark/results/jmh_benchmark.csv` folder
- double click `OPEN NOTEBOOK.sh` which starts the Jupyter lab server and should open a web browser
- to manually open the browser right click the link that should be presented in the terminal and select "open link"
- the notebook is already fully evaluated using our original data and shows the figures of the paper at the end of cells 5,8, and 10
- On the top of the notebook press the » button (hover over text is "restart the kernel, then re-run the whole notebook") to regenerate the figures using your new benchmark results

To compile the benchmarks:

- open the `encrdt source code` folder
- right click `compile benchmarks.sh` and select 'Run as a Program'
- this compiles the benchmarks.jar in the current folder wich can replace the benchmarks.jar on the desktop

The demo application can be tried by:

- double click `START TODOLIST SERVER.sh` (select "execute in terminal") to start as many intermediaries as you wish
- this should output "untrusted replica listening on: `untrusted@ws://.../`"
- double click `START TODOLIST CLIENT.sh` (select "execute in terminal") to start as many trunsted replicas as you wish
- the todo list has a large empty space at the top for the todos
- enter text in the first input box below and press the + button to add an entry to the list
- click the checkbox next to an entry to toggle its state

- at any time, connect to the an untrusted intermediary by pasting the `untrusted@ws://.../` link from the intermediary into the second box and click connect
- a replica can connect to any amount of intermediaries
- feel free to experiment with arbitrary topologies, just note that there is no persistence in our example (because that makes experimentation harder): if the last replica/intermediary that contains some information is shut down, that information is lost

To compile the todolist demo:

- open the `encrdt source code` folder
- right click `compile todolist.sh` and select "Run as a Program"
- this compiles the todolist.jar in the current folder wich can replace the todolist.jar on the desktop

4 Reusable

While our library of CRDTs is not a primary contribution of the paper, the implementations can be found, expanded, or modified. All lattice/crdt implementations can be found in a single package, the counter for for example:

- The lattice datastructure of the counter can be found in `scr/main/scala/encrdt/lattices/CounterLattice.scala`
- The operations of the counter are composed into a CRDT in `scr/main/scala/encrdt/crdts/Counter.scala`
- you will also find all other CRDT implementations in the same packages.

One can also experiment with different optimizations for encrypted CRDTs by modifying (or adding) the implementations in: `src/main/scala/encrdt/encrypted/`

Or modify the todo list application as an example how to use the library for interactive applications:

- Server: `examples/Todolist/src/main/scala/intermediaries_demo/UntrustedReplicaApp.scala`
- Client: `examples/Todolist/src/main/scala/intermediaries_demo/TrustedReplicaDemoApp.scala`

See the prior sections on details how to compile the resulting modifications.

The source code is also be more readily available as a standard code repository using an Apache 2 License at `https://github.com/ckuessner/encrdt`.

5 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `https://figshare.com/s/747c96224870d06b9b3f`, and the continued development of the process happens at: `https://github.com/rescala-lang/REScala`

6 Tested platforms

We provide the artifact as a OVA VM Image (tested to work with VirtualBox 6.1_26_ubuntu).

The VM itself is a standard Ubuntu 21.10 installation. If you wanted to recreate the VM install the following dependencies:

```
sudo apt install openjdk-17-jdk python3-pip
pip install jupyterlab "matplotlib~=3.5.0" "pandas~=1.3.4"
```

Note that the provided image is also stripped down for size reasons.

7 License

8 MD5 sum of the artifact

3ba91a4305010f655c2c90e71383e4e6

9 Size of the artifact

2.46 GiB