

# Brooke Dolny

Software Engineer

bmdolny@uwaterloo.ca  
github.com/brookedolny  
linkedin.com/in/brookedolny



## Experience

### Software Development Student · BlackBerry

Sept 2019—Dec 2019

- Developed security solutions for vehicles by integrating Cylance technologies into QNX.
- Designed and implemented a consumer–producer system for processing vehicle messages efficiently.
- Implemented a state machine for identifying a driver based on vehicle data.
- Wrote an asynchronous interface for receiving messages from a gRPC stream.

### Secure Software Developer · ECRYPT

Jan 2019—Apr 2019

- Contributed to the CysurV2X Software, developing secure vehicle to vehicle communication with C++11.
- Implemented a parallelized manager for validating the format of different types of V2X messages conforming to North American and European transportation standards.
- Identified and resolved race conditions throughout the codebase.
- Modernized C++ codebase.
- Wrote documentation for the implementation details of various classes and build processes.
- Ran retrospective meetings as part of the team's agile process.

### C++/Backend Developer · DF/Net Software

May 2018—Aug 2018

- Fixed bug in hashing function allowing over 8000 users to login after weeks of lockout; introduced better C programming practices as a result.
- Designed an API endpoint that resulted in up to 4.5x faster page load times for a web application.
- Created a report on security flaws and vulnerabilities in C, C++, and JavaScript codebases.

## Projects

### Personal Cryptographic Library

Apr 2019

- Developed a working implementation of the AES with CBC and ECB operational modes, and the SHA2 suite of cryptographic hashes, written in C.

### Fractal Generator

Apr 2018

- A fractal art generating program written in Java that supports the Mandelbrot and Julia sets.
- Leverages parallel programming to improve performance by using a worker pool.
- Utilizes the principles of OOP to improve readability.

### Traffic Simulator (Traffix)

Nov 2017

- Simulates traffic flow in an arbitrary city defined by a graph.
- A graphical interface presents to the user a real-time view of congestion and traffic patterns.
- The application was written in C++ with the Qt framework.

## Education

### Candidate for Bachelor of Software Engineering · University of Waterloo

2022

- President's Scholarship with Distinction

## Interests

Ice Hockey, Classical Music, Audio and Photo Restoration, Reading, Math, Marvel (MCU)