Ganyu Xu

xuganyu96@gmail.com | +1-604-329-5822 | linkedin.com/in/ganyu-bruce-xu | github.com/xuganyu96

Education

U of Waterloo, MASc in Electrical and Computer Engineering, GPA 94.6/100 — Expecting August 2025
Courses: public-key cryptography, applied cryptography, post-quantum cryptography

U of California, Berkeley, BA in Mathematics and Statistics, GPA: 3.464/4.0 — Sep 2015 – May 2019

Technical skills

Programming languages: Rust, Python 3, C, Bash, SQL

Cryptography & network security: post-quantum cryptography, finite-field arithmetic, TLS

Web development: Flask, Docker, AWS serverless stack, HTML/CSS, Linux, Apache Airflow, PostgreSQL/MySQL

Experience

Graduate research assistant, University of Waterloo – Waterloo, Ontario, Canada — May, 2024 - Present

- Designed and implemented a post-quantum key encapsulation mechanism based on ML-KEM with an alternative CCA transformation, achieving IND-1CCA security for ephemeral key exchange (e.g., TLS handshake), optimizing decapsulation speed by up to 70%, and reducing round-trip time by up to 50%.

Senior data engineer, LeanTaaS Inc. – Santa Clara, CA, United States — July 2019 - Sep 2023

- Overhauled nightly data ingestion and preprocessing pipeline: migrated CRON jobs on a single EC2 server to an **Apache Airflow** cluster on **AWS Elastic Container Service**
- Designed and implemented serverless and cost-effective data warehouse. Raw data is parsed and written to AWS S3 in columnar format (**Apache Parquet**), then queried using distributed SQL query engine **AWS Athena**
- Improved data pipeline observability by using **DataDog** for service health monitoring and log aggregation; shortened production outage response time by connecting DataDog health checks with **PagerDuty**
- Mentored junior developers through project design review, pair programming, and technical workshops

Projects

RustCrypto — github.com/RustCrypto

RustCrypto is a collection of common cryptographic primitives implemented in pure Rust.

In November 2023, *a timing variability side channel* was discovered for major RSA implementations including **RustCrypto/RSA**. I participated in the effort to mitigate this side channel vulnerability by migrating dependency on generic big integer library to constant-time big integer library **RustCrypto/crypto-bigint**. Specifically:

- Ported stack-only random prime generation to heap-allocated big integers
- Integrated Marvin toolkit for detecting timing variability in RSA implementation

rustls — github.com/rustls/rustls

- Identified incorrect TLS termination at google.com and redirected the example binaries to connect to other correct TLS server implementations

Apache Airflow — github.com/apache/airflow

- `AirflowFargateExecutor`: an alternative task executor that launches one ephemeral worker container per task instance. Ephemeral workers save users from paying for idle worker when task load is low. Hosting workers on serverless container orchestration like AWS Fargate simplifies massive task parallelism
- Fixed improper exception handling in python multiprocessing that causes critical process to become zombie instead of exiting upon database disconnect
- Fixed data structure serialization that causes frontend to crash when said data structure cannot be serialized into JSON
- Implemented user-specified database check retries in the CLI command `airflow db check`