# Ke Li

(780)-885-0852 | kegrad2023@gmail.com | Toronto, ON, Canada
damianli.com | Linkedin

## Education

**University of Toronto** Toronto, ON

*Master of Engineering in Computer Engineering & Identity, Privacy and Security (IPS) Jan. 2024 – Dec 2025*

- Coursework: Computer Security, Cloud Computing, Deep Learning & Neural Network
- GPA: 4.0/4.0

**University of Alberta** Edmonton, AB

*Bachelor of Science in Computer Science* *Sep. 2019 – Aug 2023*

- Coursework: Operating System, Computer Networks, Computer Architecture, Web Development, Mobile App Development, Database Management, Machine Learning, Agile Methodology
- Awards: *Dean's Honor Roll (22-23)*

## Experience

**Student Developer** Jan. 2023 – Apr. 2023

*University of Alberta ALT Lab* *Edmonton, AB*

- Developed a website that generates interactive word graphs based on Cree words and their domains using React, D3.js and Docker.
- Optimize graph load times from 10 to 0.5 seconds by implementing the Lazy loading time Design Pattern.
- Reviewed 50+ PR and contributed 4K+ lines of code to the codebase via Git.
- Maintained web application documentation using MkDocs.
- Utilized Jest for unit testing and Cypress for E2E testing.

## Projects

**Distributed Linux Performance Analysis and Monitoring System** May. 2024 – Jul. 2024

- Developed a Docker-based setup to build project environments with dependencies, facilitating easy deployment across multiple servers.
- Implemented the monitor module using the Factory Design Pattern to create an abstract monitoring interface, including CPU status, system load, software interrupts, memory, and network monitoring.
- Built a Distributed System using gRPC; Deploying server on target machines and client library used by monitor and display modules, ensuring low coupling and high modularity.
- Utilized Protobuf for serialization to define comprehensive data structures for the project.
- Constructed the display module by using Qt for UI design and extending QAbstractTableModel for data modeling.
- **Utilized**: C++, CMake, Docker, gRPC, protobuf, Qt

**SDN Performance Simulator** Screencast Mar. 2024 – Apr. 2024

- Developed a web-based tool that allows users to create custom SDN network topologies and analyze the performance of their simulated networks.
- Implemented (CI/CD) pipelines to streamline the development process, enable efficient code deployment.
- Enhanced security of data fetching by integrating AWS Lambda and API Gateway, ensuring secure HTTPS connections for back-end server communications.
- **Utilized**: Python, JavaScript, React, Django, Mininet, Docker, AWS EC2, REST API

**Computer Security CTF** Blog Jan. 2024 – Apr. 2024

- Network Vulnerability: | DNS Poisoning | XSS Attack | CSRF Attack | SQL Injection |
- Binary Exploitation: | Buffer Overflow | Format String Attack | Double free Attack | ROP |
- **Utilized**: C++, C, GDB, Linux, Python, JavaScript, HTML

## Technical Skills

**Languages**: C/C++, Python, Golang, Java, SQL (Postgres), JavaScript, HTML/CSS
**Frameworks**: React, Django, Cypress, Jest, GraphQL, Material-UI, Ant-Design
**Developer Tools**: CMake, gRPC, protobuf, Qt, Git, Docker, Kubernetes, AWS
**Compiler Tools**: LLVM, Flex, Bison
**Libraries**: Scikit-learn, Pandas, NumPy, Matplotlib, Scapy, D3.js, ReactFlow, Mininet