

Shijie(Freeman) Zhang

☎ (410)736-9987
✉ ShijieZhang@jhu.edu

🏠 2033 Manhattan Avenue 21, East Palo Alto, CA 94303
🔗 <https://www.linkedin.com/in/FreemanZhang>

EDUCATION

Johns Hopkins University Master of Science in Information Security	<i>September 2013 - May 2015</i> Major GPA:3.85/4.00
University of Science and Technology of China Bachelor of Engineering in Computer Science	<i>August 2009 - July 2013</i> Major GPA:3.51/4.00

SKILLS

Languages: Proficient(Java, Javascript), Prior Experience(C, Python, HTML5/CSS, Coffeescript, PHP, Ruby, SQL)
Web: Node.js, Express, Rails, Ajax, jQuery, REST, Bootstrap, Spring

WORK EXPERIENCE

Software Development Intern for an AWS Role Permissions Compliance System (Python, AWS and Splunk)
Financial Industry Regulatory Authority (FINRA), Rockville, MD *May 2014 - August 2014*

- The system featured in *FINRA's* presentations in *Splunkconf. 2014* and *AWS re:Invent 2014*
- Automated the process of converting role permission policies from excel files to JSON formats
- Imported AWS CloudTrail data into Splunk to monitor permission changes of AWS IAM APIs
- Developed Python scripts to check the compliance of AWS APIs and defined the severity of security incidents

Research Assistant for Multimedia Processing and Information Security (Matlab, Latex, OpenGL)
National Engineering Lab of Speech and Language Information Processing, Hefei, China *March 2012 - July 2013*

- Journal Paper "Automatic Tag Saliency Ranking for Stereo Images"
Yang Cao, Kai Kang, **Shijie Zhang**, et al. Accepted in *Neurocomputing, 2015. (5-Year IF: 2.102)*
- Conference Paper "Underwater Stereo Image Enhancing Using a New Physical Model"
Shijie Zhang, Jing Zhang, et al. *21st International Conference on Image Processing (ICIP), 2014.*
- Journal & Conference Paper "A Novel Segmentation-based Video Denoising Method with Noise Level Estimation"
Shijie Zhang, Yang Cao, et al. *19th International Conference on Multimedia Modeling (MMM), 2013. Oral.*
Also invited at *Information Sciences*, Vol. 281, pp. 507-520, October 2014. **(5-Year IF: 3.969)**

PROJECT EXPERIENCE

🔧 A Scalable Instant Messaging Single-page App (Express, Node.js, jQuery, MongoDB)

- *Client-side routing*: Created the UI layout and associated history control supports
- *Modular Javascript*: Used the module pattern and publish/subscribe pattern to develop manageable units
- *Server-side synchronization*: Developed full-duplex communication channels in with Socket.IO and JSON
- *NoSQL*: Validated incoming client data and implemented MongoDB CRUD operations

🔒 A Secure Web-based Group Forum (PHP, Bootstrap, MySQL)

- *Data structure*: Designed an efficient database schema to store posts and its corresponding replies
- *Data filtering*: Performed validation and sanitation to eliminate XSS, CSRF and SQL Injection vulnerabilities
- *Access control*: Restricted access to members-only pages by authentication, authorization and session control
- *Availability*: Implemented automatic MySQL database replication to recover from server failures

🔧 A RESTful Online Poll Web Service (Java, JPA, REST)

- *Resource modeling*: Identified and developed REST resources, endpoints, actions and responses
- *Error handling*: Validated API inputs and implemented meaningful error response messages
- *Paging and versioning*: Employed pagination to handle large datasets and URL versioning to handle API changes

🔧 A High Performance E-commerce Website (Ruby on Rails, Coffeescript)

- *AJAX and internationalization*: Ajaxified the shopping cart and created translation files for alternate locales
- *Memory optimization*: Profiled memory to locate choke points and implemented according ROR best practices
- *Caching*: Developed various caching strategies (ETag, view caching, data caching and Ajax caching)

🔧 A Fault-tolerant Distributed Chat Room Application (C, SpreadToolkit)

- *Availability*: Deployed Spread daemons on different servers and redirected clients to available servers when necessary
- *Consistency*: Built an efficient UDP-based multicasting program based on Lamport timestamps for synchronization
- *Failure recovery*: Implemented reconciliation algorithm to recover from network partitions and server crashes