Avraham "Abe" Bernstein | CV

Author: Avraham "Abe" Bernstein
Email: Avraham DOT Bernstein PLUS cv AT gmail DOT com
Tel/Whatsapp: +972.54.641-0955
City: Jerusalem 9727433 ISRAEL
Time Zone: UTC +02:00/+03:00 (winter/summer)
Shabbat Observant: Not available Fri. evenings (Jm time) beginning 1 hour before sunset until Sat. night 1 hour after sunset, nor on Jewish holidays
WWW: https://www.avrahambernstein.com
Linkedin: https://www.linkedin.com/in/avrahambernstein/
Last Update: 2025-05-22

Introduction: Cr8tive Solutions To Hard Problems

1. I have more than 40 years experience in state-of-the-art software development:
   - inventing new algorithms
   - inventing domain specific languages (DSL) which can grossly simplify many difficult problem domains, especially for non-programmers
   - using compiler tools to automatically refactor source code, and for anti-reverse engineering obfuscation
2. I am a "hands-on" master software engineer. I enjoy programming. I design prototypes and minimum viable products for *CTO* groups.
3. I have many inventions and some **recent patents** in a wide variety of application domains, e.g.:
   - digital automotive technology
   - Internet TV
   - cybersecurity
   - bioinformatics
   - accessibility
   - factory automation
   - VLSI CPU design
4. I am an expert generalist and an autodidact polymath[1]. I easily learn and become an expert in new fields.
5. In order to jump start my learning process I prefer to find a domain expert to mentor me, and I am also a good mentor myself.
6. I know many computer languages which I am able to learn easily due to my compiler background.
7. My "go to" *programming* languages in which I am quite proficient are the following:
   - C
   - Python and Beautiful Soup
   - Jinja2, the "gold standard" macro and template preprocessor
   - Pyexpander, which is much simpler to use than *Jinja2* and just as useful *except* where template inheritance is needed, *but* unfortunately is *polluted* by a GPLv3 license
   - bash, including Posix CLI commands

8. My "go to" *data* languages are *HTML*, *Markdown*, pandoc, srcML, *XML*, *YAML*, and *Excel*.
9. I thrive on undertaking new challenging projects. I am quite comfortable engaging in and leading *flexible* "brain storming" sessions.
10. I have a long history of providing considerable value added to my employers. Browse the links below (typically preceded by the *right pointing triangle* symbol "▶") that describe the details of my career. They will reveal a long history of "home runs". I am especially proud of my inventions in the fields of bioinformatics and accessibility (and here) even though they are not in the application domains where I usually work.

Work Experience

2025-present: Independent Commercial S/W Cybersecurity Obfuscation Product Development

I am starting (2025-06-01) my own firm to develop commercial obfuscating (= anti reverse-engineering) compilers for:

1. *C11/C++2014*
   - implementation via srcML (commercial license) and Python's Beautiful Soup (MIT license)
2. web assembly (WASM)
   - implementation via the parsimonious (MIT license) Python PEG parser

And I intend to develop lightweight and inexpensive commercial compiler utilities that refactor *C* source code, via *srcML* and *Beautiful Soup*, including:

1. enabling optional and named function call arguments
   - implementation via the GCC designated initializer extension
2. obfuscating name mangler
   - similar to stunnix cxx-obfus
3. compile time *constant* string encryption that decrypts at startup
4. *static* variable and function name obfuscation
5. fully functional ELF dynamic shared object (DSO) files that shroud their global export table
6. detection and elimination of duplicate code where variable names are parameterized
   - implementation via the GCC statement expressions extension, and the clang block extension

Obfuscation Explanation

A professional software engineer who is an experienced "hacker" is usually able to "reverse engineer" well designed software even when the software is delivered as object code via a compiler; but P-code such as that generated by Java is trivial to hack. Obfuscation mixes up the software as if it were processed with a blender. Of course correct obfuscation must retain the original code logic, but its code footprint can only be slightly larger than the original and execute slightly slower, otherwise the obfuscation has no commercial value. The obfuscation places "speed bumps" in front of the hacker, but eventually, say within a few weeks or months, a good hacker will overcome the obfuscation; therefore it is incumbent upon the

obfuscater to regularly supply another randomized version every few months. For example the kind of software that contains secrets that should be obfuscated includes unexploded military ordinance, media players of proprietary content, banking/bitcoin applications, etc.

2022-25: Aurora Labs Tel Aviv IL

▸ Automotive Software Updates: CTO Group: (a) Invented a patent-pending algorithm to greatly reduce RAM consumption during FLASH updates which improves compression efficiency which is the core KPI of the software update industry. (b) Greatly improved the CPU efficiency and RAM consumption of the company's core product that automatically refactors embedded C source code of their automotive industry clients.

2021: Morphisec Beer Sheva IL

▸ Server Cybersecurity: Reverse engineering and refactoring of X64 object code

2021: consultant to Qedit Tel Aviv IL

▸ Banking Cryptographic Algorithms: Securing C algorithms in a web browser using WASM

2018-20: Argus Cyber Security Tel Aviv IL (restructured as PlaxidityX IL, subsidiary of Elektrobit/Continental DE)

▸ Automotive Software Updates: (a) Patented an algorithm greatly reducing FLASH memory required to implement mini-bsdiff. (b) Architected and implemented an embedded S/W update driver based upon mini-bsdiff and the xz binary compressor.

2016 part-time: Canary Mission Jerusalem IL

▸ Consultant: SOHO Cybersecurity "Hygiene"

2014-17: Viaccess-Orca Ra'anana IL (subsidiary of Orange FR)

▸ Internet TV Infrastructure: Cybersecurity obfuscation manager

2013-14: *Discretix* Netanya IL (renamed *Sansa Security*, acquired by ARM UK)

The company sold their Internet TV business unit to Viaccess-Orca (2014) above.

2013 part-time: *NVT* (defunct) US

▸ CTO Nigerian Agritech

2012: Telequest (stealth) Jerusalem IL

▸ Vehicle Navigation Algorithms: VP R&D: Traffic jam reduction algorithms

2011: consultant to <u>Syntezza Bioscience</u> Jerusalem IL

▸ <u>PCR MRSA Kit: Inventor of bioinformatic PCR algorithms using AI threshold technique, and more accurate bioassay normalization and noise reduction</u>

2005-10: *NDS* Jerusalem IL (acquired by <u>Synamedia</u> UK)

▸ <u>Internet TV Infrastructure: Cybersecurity researcher</u>

2004: *Vyyo* (defunct) Jerusalem IL

▸ <u>Broadband RF Networking: Architect of super-efficient cable modem testing laboratory</u>

2002-03: *Virtouch* (defunct) Jerusalem IL

▸ <u>Blind Accessibility Device: VP R&D: Inventor of product that allowed the blind to see/understand images in a PC/smartphone web browser ideally combined with a consumer grade graphics tablet</u>

2002: *TMT* (defunct) Jerusalem IL; Local Area RF Networking

I did similar tasks for *Vyyo*. See <u>Vyyo (2004) above</u> and <u>Vyyo (2000) below</u>.

2002 part-time: *Jolt* Jerusalem IL (acquired by *MRV Communications* IL, and eventually by <u>Adtran</u> US)

▸ <u>Free Space Optics Networking: Consultant: Designer of SNMP NMS client and agent</u>

2000-02: *Vyyo* (defunct) Jerusalem IL

▸ <u>Broadband RF Networking: Manager of S/W utilities group; inventor of cable modem hybrid (RF/dial-up) IP allocation protocol</u>

1999: contractor to *Phasecom* Jerusalem IL (acquired by *Vyyo* above)

Broadband RF Networking: See the tasks that I did for the successor company <u>Vyyo (above)</u>.

1998: contractor to *Fourfold* (defunct) Jerusalem IL

▸ <u>Fabless VLSI CPU Design: Novel GCC compiler port for a FORTH-like CPU</u>

1996-97: CEO *Pitkha* (defunct) Jerusalem IL, contractor to *Optimet* Jerusalem IL, subsidiary of <u>Ophir Optronics</u> IL

▸ <u>Conoscopic Laser Interferometry: S/W architect of a DSL for a 2D measurement robot</u>

1996: CEO *Pitkha* (defunct) Jerusalem IL, contractor to [Elop/Elbit](#) Rehovot IL

▸ Military Optical Devices: S/W architect of a DSL to implement a mil-spec automated testing laboratory for the BlackHawk helicopter weapons targeting system

1992-95: CEO *Pitkha* (defunct) Jerusalem IL, contractor to [DSP Group](#) Ramat Gan IL

▸ Fabless DSP CPU Design Center: Inventor and S/W architect of a DSL to implement the software tool chain for the PINE CPU

1990-91: contractor to [Digital Equipment Corp (DEC)](#) Herzliya IL (eventually acquired by [Hewlett-Packard](#) IL), contractor to [Iscar Metalworking](#)

▸ Metal Blade Production Factory: Co-inventor and S/W architect of a DSL to implement a shop floor production control system that orchestrated a completely automated factory

1988-89: contractor to *Cubital* (defunct) Herzliya IL, subsidiary of *Scitex* IL (acquired by [Hewlett-Packard](#) IL)

▸ One Of The Original 3D Printers: S/W R&D

1989: contractor to *Cubital* (defunct) Herzliya IL, subsidiary of *Scitex* IL (acquired by [Hewlett-Packard](#) IL)

▸ PC Accessibility Device For Quadriplegics: Inventor and S/W architect

1987: *Orisol* (defunct) Lod IL

▸ High Speed Sewing Robot For Leather Goods: S/W architect of a DSL used to control the robot

1980-86: Junior Programmer and Economist Positions in the US and Israel

1977: [Ontario Energy Board](#) Toronto Ontario

▸ Public Utility Commission: Public interest intervenor-economist at the ECAP77 hearings on marginal cost pricing for electricity

Unpatented Personal Inventions

▸ Inventions

Part-time Teaching Experience

▸ Teaching

Education

1. **1978-79:** <u>York Univ Graduate School</u> Toronto Ontario: **masters degree in economics with a minor in applied mathematics**[2]

   - My major project was an economic-engineering simulation of a hydro electric dam in *FORTRAN*.

2. **1976-77:** <u>Univ Of Toronto Rotman Graduate School Of Management</u> Toronto Ontario: **no degree, applied credits to York Univ (above)**

3. **1973-76:** <u>Univ Of Toronto, Undergraduate School Of Arts & Sciences</u> Toronto Ontario: **BA economics**

4. **1969-73:** Secondary School, Ontario Canada: **"Honours" (grade 13) high school diploma**

   - I took my first course in computer science in *FORTRAN* on an <u>IBM 1130</u> mini-computer with 16 KB RAM. My first serious program was a *perfect* game of *Qubic*, 3D tic-tac-toe, using 4 levels of boards each of which has 4x4 squares. Subsequently I became "addicted" to programming for life.

   - Also besides advanced courses in mathematics, one of my more useful courses was learning how to "touch type".

Personal

▸ <u>Personal</u>

Colophon

▸ <u>Colophon</u>

Footnotes

---

1. **Autodidact Polymath:** There is a common misconception that only geniuses like *Leonardo da Vinci* deserve the appellation autodidact polymath, and therefore by referring to myself as one then I am making the bombastic claim to be a genius in the same category as Leonardo. There is an outstanding <u>TED talk</u> showing how children from Indian slums with the appropriate mentoring can become *autodidact polymaths*. And there was a similarly successful project done with children from <u>Mexican slums</u>. Many very intelligent people, especially as they get older, don't like moving outside of their intellectual comfort zone, whereas I revel in taking on intellectual challenges in new fields outside of my comfort zone. Note my "grand slams" in a wide range of application domains, where the only way to quickly achieve expertise was self-learning admittedly with the help of highly qualified mentors. I feel that I still maintain a high degree of child-like <u>mental plasticity</u>. Part of this skill I retain by regular interaction with my own young grandchildren, where my play with them is much more analytical than I used to exhibit with my own children.↩

2. **York Univ:** Exceptionally I passed my comprehensive examinations before I took any of the required economics courses! Therefore the school allowed me to take any accredited courses from any Canadian university. I had initially wanted to take a graduate degree in engineering from the Univ of Toronto but they refused to accept me as a *regular* student, but they allowed me to enroll as a *special* student. In fact most of my courses for the York degree were from the Univ of Toronto graduate school of engineering.↩