# Avraham “Abe” Bernstein | Master S/W Engineer | Abbrev CV

**Under contract to TK Open Systems Ltd.**

*Version: 0.1-tkos-abbrev*

*Last update: 2017-09-18T17:30:22Z*

## 0.1 Contact Info & Links

**email/skype:** Avraham.Bernstein@gmail.com

**geolocation:** Jerusalem Israel, **tz:** UTC +02:00/+03:00 [winter/summer]

**tel-IL-mobile/whatsapp:** +972.54.641-0955

**cv-full-html:** http://purl.org/Avraham.Bernstein/tkos/cv.html

**cv-abbrev-html:** http://purl.org/Avraham.Bernstein/tkos/cv-abbrev.html

**cv-full-docx:** http://purl.org/Avraham.Bernstein/tkos/cv.docx

**cv-abbrev-docx:** http://purl.org/Avraham.Bernstein/tkos/cv-abbrev.html **[this file]**

# 1.0 Summary

I am an experienced computer scientist and S/W architect. I have devised innovative solutions to many S/W problems for a wide range of fields, including cybersecurity, cryptography, bioinformatics, factory automation, VLSI CPU design, telecommunications, blind vision, accessibility, transportation vehicle route guidance, and test automation. I have worked for a number of organizations, large and small, and helped them realize improvements in their product performance, often putting them in the front rank in their field. I have acquired expert knowledge in a number of fields, often liaising with noted experts, and have been able to quickly apply this knowledge to improve the competitive position of the companies and their products. I have a keen interest in computer languages, both practical and theoretical. I have created a number of domain specific languages (DSL) that were instrumental in greatly simplifying seemingly intractable problems.

# 2.0 Work Experience

### 2017 - present: Cybersecurity Consultant

*@Self-Employed, Jerusalem:*

Keys: cybersecurity, architect, algorithms, obfuscation, compiler, **C/C++**, javascript, WASM

1. I am developing an obfuscating compiler for C/C++ and for Web Assembly (WASM). Still in stealth mode.
2. I am a security mentor for the Jerusalem Mass Challenge start-up hub.

### 2011-17: S/W Architect & Developer: Cybersecurity: OTT Internet Pay TV System

*@Viaccess-Orca, Ra’anana - a subsidiary of Orange FR, and @Discretix/SansaSecurity, Netanya - now merged into ARM:*

Keys: cybersecurity, DRM, architect, algorithms, anti-reverse engineering, obfuscation, LLVM compiler, cryptography, **C/C++**, bash, Python, TCL, Android, IOS, Linux, ELF

1. I architected and implemented anti-reverse engineering and obfuscation programming frameworks and libraries in C/C++ for their DRM protected movie player application that ran on Android and IOS devices.
2. The challenges of implementing obfuscation are that (1) the other programmers should not be concerned about it because their focus must be on writing correct code, and (2) the resulting increase in size and reduction in run-time speed must not noticeably reduce the usability/functionality of the application. In general the aim of obfuscation is to provide “good enough security” that will deter 95% of potential attackers, and when combined with regular application updates will force an attacker to begin his next reverse engineering attempt from scratch.
3. I developed a post processor to obfuscate the resulting binary object ELF files.
4. I developed a light weight obfuscated cryptographic library implemented as a H file using inline functions so that every module that included it had its own private copy of the library with a module specific randomized implementation which prevented an attack against a single core cryptographic module that could potentially subvert the whole application.
5. I developed an Android root detection mechanism using fuzzy logic techniques.
6. I developed a background watchdog security thread to dynamically ensure that the binary code had not been tampered with.
7. All secure code modules were implemented as native libraries written in C/C++.
8. Offline utilities and build scripts were written in bash, Python, and TCL.
9. My typical development methodology was to first build a prototype for desktop Linux, secondly as a standalone CLI application on the target device, and finally to incorporate the source code into the full application on the target device. Whenever possible I preferred to test on virtual machines.
10. I was responsible for the purchase decisions and usage policy of 3rd party obfuscation and cryptographic utilities and libraries.
11. **At the end of my 6 year tenure there were 40M subscribers, and no security breaches.**

### 2016-16: Cybersecurity Consultant: Protection of a Small Business with Extremely High security Concerns

*@Anonymous, Jerusalem:* See details.

Keys: cybersecurity, privacy, anonymity, WordPress, static web site, Cloudflare, Windows, Android, Google Docs, Google Drive

### 2010-11: S/W Architect & Developer: Transportation: Urban Traffic Vehicle Route Guidance Algorithms

*@TeleQuest, Jerusalem:* See details.

Keys: urban vehicle route guidance, architect, algorithms, Java, AWS

### 2009-09: S/W Architect & Developer: Bioinformatics: PCR algorithm

*@Syntezza, Jerusalem:* See details.

Keys: bioinformatics, PCR, architect, algorithms, mathematical programming, **C**, Python

### 2004-09: Cybersecurity Researcher for a CA Satellite Pay TV System

*@Cisco-NDS, Jerusalem:* See details.

Keys: cybersecurity, DRM, algorithms, cryptography, anti-reverse engineering, obfuscation, LLVM compiler, VM, QEMU, RPC, automated testing, S/W quality, **C/C++**, TCL, Python, Linux, bash, Win32

### 2002-03: S/W Architect & Developer: Accessibility: Enabled Blind to "See" Maps

*@Virtouch, Jerusalem:* See details.

Keys: accessibility, blind, architect, GIS, MapML, HTML, SVG, javascript, XSLT, XML Schema, XSLT, **C**, TCL

### 1999-2002: S/W Architect & Developer: Network: Utilities for a "Wireless" Cable Modem and Router System

*@Vyyo, Jerusalem:* See details.

Keys: network, architect, algorithms, SNMP, SNMP-agent, NMS, automated testing, **C**, TCL, embedded

### 2001-01: S/W Architect & Developer: Network: Network Management System (NMS) for a FSO Device

*@MRV-Jolt, Jerusalem:* See details.

Keys: network, architect, SNMP, SNMP-agent, NMS, Java, **C**, TCL

### 2001-01: Consultant: Network Management System (NMS) for a Cable Modem & Gateway System

*@Foxcom - One Path Networks, Jerusalem:* See details.

Keys: SNMP, NMS

### 2000-01: S/W Developer: Communications: Win32 Asynchronous TCP/IP DLL for a Visual Basic Project

*@Inex-Zamir, Jerusalem:* See details.

Keys: TCP/IP communications, **C**, Visual Basic, Win32, soft real-time

### 1998-99: S/W Architect & Developer: Compiler: GCC Compiler Port for a 128-Core Stack Machine

*@Fourfold, Jerusalem:* See details.

Keys: C compiler, gcc, architect, FORTH, **C/C++**, TCL, LISP

### 1997-98: S/W Architect & Developer: Factory Automation: Conoscopic Interferometer Workstation

*@Newport-Optimet, Jerusalem:* See details.

Keys: measurement workstation, architect, DSL, **C**, TCL, OpenGL, Win32, soft real-time

### 1996-97: Consulting S/W Engineer: Lecturer for a Win32 Internals Course

*@M.E.R., Jerusalem:*

Keys: lecturer, Win32

### 1996-96: Consulting S/W Engineer: Win32 Techniques to Improve Performance of a Soft Real-Time Biofeedback Application

*@Ultramind, Jerusalem:*

Keys: Win32, soft real-time

### 1996-96: Consulting S/W Engineer: Design of Win32 Device Driver for a Frame Grabber

*@Cefar, Jerusalem:*

Keys: Win32, soft real-time

### 1995-96: S/W Architect & Developer: US DOD Mil-Spec Automated Testing: Night Hawk Fire Control System

*@Elbit-Elop, Rechovot:* See details.

Keys: automated testing, mil-spec, architect, DSL, **C/C++**, BASIC compiler, lex/yacc, Win32, soft real-time

### 1991-94: S/W Architect & Developer: VLSI: Simulator & S/W Toolchain For DSPG PINE CPU

*@DSPG, Givat Shmuel:* See details.

Keys: VLSI simulator, S/W Development Toochain, architect, DSL, **C/C++**, lex/yacc, assembly, Win32

### 1989-91: S/W Architect & Developer: Factory Automation: Shop Floor Production Control (SFPC) System: BARI II

*@Digital Equipment Corporation (DEC), Herzliya, for @Iscar, Tefen:* See details.

Keys: factory automation SFPC, architect, DSL, Pascal, SQL, VAX/VMS

### 1988-88: S/W Architect & Developer: Accessibility: Quadriplegic PC Accessibility

*@Cubital, Herzliya - a charity project funded by the company and their CEO Itzhak Pomerantz in cooperation with the Beit Levinson Rehabilitation Hospital, and the IDF Rehabilitation Unit:* See details.

Keys: accessibility, Prolog, PC-DOS

### 1987-88: S/W Developer & VAX/VMS Sysadmin: 3D Printer: Solider

*@Cubital, Herzliya:* See details.

Keys: 3D printing, **C**, sysadmin, VAX/VMS

### 1986-87: S/W Developer: Soft Real-Time RS232 Z80 Communication Driver: Data Collection & Access Control Terminal

*@Elde, Jerusalem:*

Keys: data collection terminal, **C**, RS232, Z80, embedded, real-time

### 1985-86: S/W Developer: Factory Automation: Leather Sewing Workstation

*@Orisol, Lod:* See details.

Keys: sewing workstation, DSL, AutoCad, **C**, awk, PC-DOS

### 1984-85: S/W Developer & VAX/VMS Sysadmin: Hebrew/English Word Processor: Glyph

*@John Bryce, Jerusalem.*

Keys: word processor, **C**, sysadmin, VAX/VMS

### 1983-84: S/W Developer: Real-Time: Data Collection Terminal & Lavi Fighter Plane Radar

*@Elta/IAI via DSI, Ashdod:* See details.

Keys: data collection terminal, PL/M, 8080, RTOS, fighter plane radar, Jovial, embedded, real-time

### 1981-83: S/W Developer & IBM CP/CMS Assistant Sysadmin

*@Mitre Corp, McLean VA:* See details.

Keys: APL, PL/1, sysadmin, IBM CP/CMS

### 1979-80: Programmer & Economist

*@JWWA.com, an economic consulting firm in the Washington DC area:* See details.

Keys: electric utility economics, Fortran, IBM MVS

### 1977-78: Intervenor/Economist

*@Ontario Energy Board (OEB), Toronto:* See details.

Keys: electric utility economics

# 4.0 Education

## 4.1 Formal Education

### 1979: York University, Canada: MA Economics & Applied Mathematics

See details.

### 1977: University of Toronto - Rotman School of Management (MBA Program): No Degree

See details.

### 1976: University of Toronto: BA Economics & Applied Mathematics

See details.

## 4.2 Continuing Education

1. Today the field of computer science is changing so rapidly, that without ongoing self-study, one’s formal education becomes obsolete within 5 years.
2. From 1991-96, pre-Internet, I used to spend one afternoon per week reading at the Hebrew University Jerusalem (HUJI) computer science library.

3. Afterwards with the advent of the Internet, more up-to-date computer science topics were available on the Internet, so going to the library was no longer the most efficient way to keep updated.
4. Since 2005, I have maintained a subscription to the O'Reilly Safari on-line tech library.
5. My daily dose of tech news comes from Slashdot.
6. I regularly watch TedX and Talks At Google video seminars.
7. The most fascinating feature of TedX talks is to watch and learn how world class experts in a wide range of fields are able to distill their special area of knowledge to intelligent laymen in just 18 minutes. Whenever I make a presentation, I attempt to emulate the best TedX speakers. Also I attempt to write presentations which emulate this TedX Art of Innovation Top 10 Format.
8. I regularly read the tech sections of the Israeli business newspapers Globes and The Times of Israel.
9. I have ecclectic interests.
10. I regularly research new topics in depth.
11. My browser bookmarks are my most important professional store of my knowledge. I use Firefox because it has the best built-in bookmarking feature, because it uses tags/labels. I have a well honed tag taxonomy.
12. I am an expert generalist, and an autodidact polymath, i.e. a self-learner in new fields who achieves expertise quickly.

## 5.0 Spoken Languages

1. English (5/5)
2. Hebrew (4/5)
3. French (2/5)

## 6.0 Computer Languages, SDKs, and Operating Systems

Language knowledge in order of expertise, based upon my current frequency of usage:

1. C, TCL, bash + posix text utilities, e.g. awk, sed, etc.
2. C++, python, make, html5, css, markdown, pandoc, jinja2
3. flex, bison, llvm, javascript, java, yaml, json, go
4. forth, lisp, prolog, apl, fortran, opengl, svg, xml schema, relax ng, xslt, perl, C#

Note that I write compilers and Domain Specific Languages (DSL), so learning a new language takes me only a few days.

O/S knowledge in order of expertise, based upon my current frequency of usage:

1. Linux
2. Android
3. Win32
4. IOS

## 7.0 Patents Under Development

1. Bioinformatics: (a) An extremely accurate and simple noise reduction and normalization algorithm to improve the accuracy of the standard PCR Ct calculation, and (b) an Artificial Intelligence (AI) methodology for measuring the quantity of DNA in a bioassay where inhibition makes it impossible to estimate the Ct because no underlying logistic function (= a flat "S" shaped curve) exists.

2. Cryptography: A set of non-linear cryptographic primitives using Hamming weight-like data dependent permutations which overcomes the well known limitation of using Hamming weights because they have a binomial distribution.

## 8.0 Personal

I was born in Canada in 1956. I have lived in Jerusalem Israel since 1983. I am married with 4 children, 2B + 2G, plus many grandchildren. I take physical fitness seriously. Once upon a time I was a judoka, and a classical guitarist. I was an IDF reserve soldier for 15 years, where I served as a combat soldier in the infantry in the Jordan Valley. In spite of the fact that I joined the army when I was 32 years old (Hebrew: *Shlav Betnik*), functionally, but unofficially, I served in the capacity of deputy company commander (Hebrew: *Samech Mem Pe*) which provided me with the opportunity to achieve rich personal growth, and enabled me to learn important managerial and leadership skills.

## Colophon

- **Generator:** This document was generated using the Pandoc universal document converter extended Markdown engine, along with the Jinja2 macro/template preprocessor. See the source code at my github site.
- **Safety & non-annoyment pledge:** This document is free of scripts, frames, ads, and animations.