
# Open Source crypto / SSL

# &

# Government accreditations

# Me (Paul Bakker)

IT

Security

Cryptography

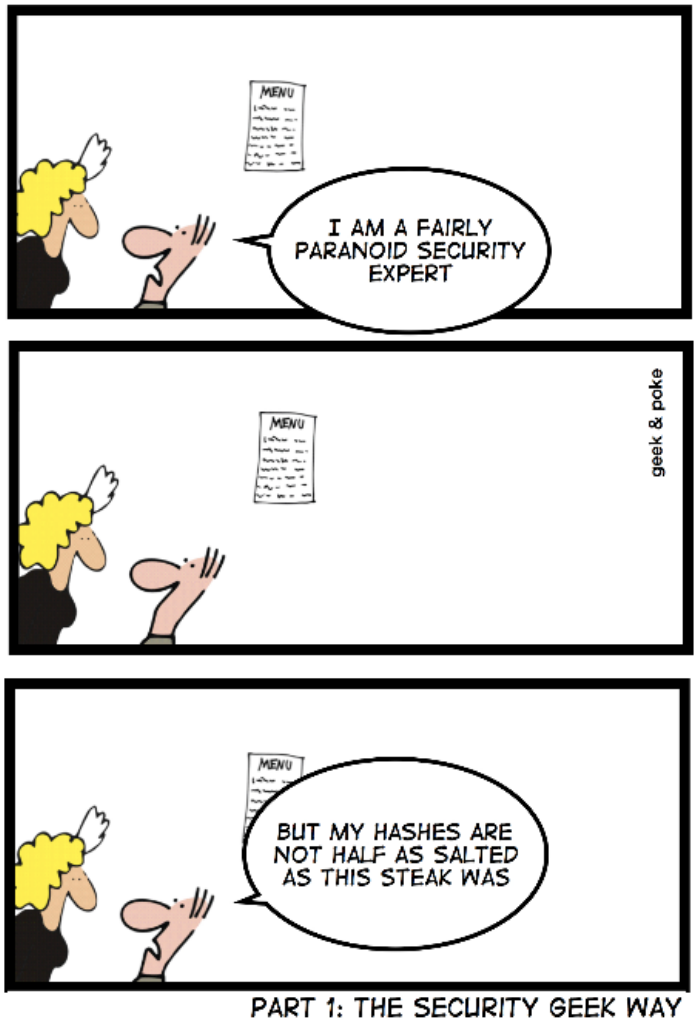
Software developer

Angel investor

@PaulBakkerNL

POLAR SSL

# 1000 WAYS TO COMPLAIN IN A RESTAURANT




PART 1: THE SECURITY GEEK WAY

POLAR SSL

# PolarSSL

Cryptography and SSL / TLS library in C

Started on 01-01-2009 as successor to XySSL

GPL / Dual Licensed

Small: 17 kSLOC, RAM 30-150 Kb, ROM 30-120 Kb

(Examples 6 kSLOC, tests 82 kSLOC (generated))

# Some projects using PolarSSL

Hiawatha webserver

RTMPDump

POWERDNS

OPENVPN™

(OpenVPN-NL)

Gnuk

uMurmur

POLAR SSL

# Availability

In repository:

- Debian / Ubuntu
- FreeBSD
- Fedore
- OpenWRT

http://polarssl.org

# Runs on

## Operating systems:

*NIX / Linux

Windows

Mac OSX

Android

iOS

ThreadX

FreeRTOS

XBox

eCOS

and more

## Architectures:

X86 / x64

ARM

PowerPC

MIPS

Motorola 6800

and more

# Philosophy

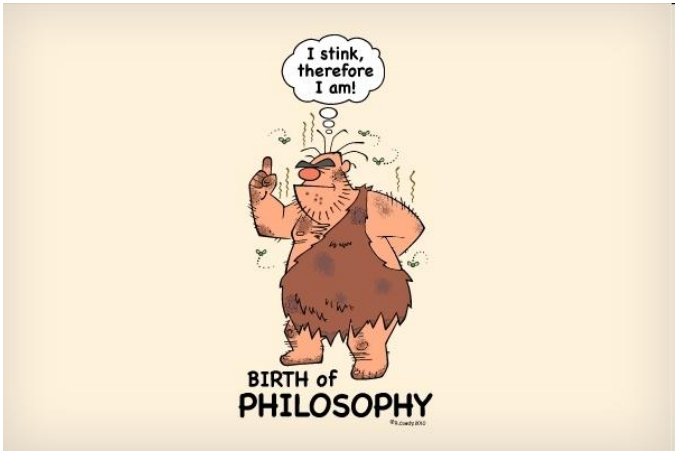
Easy

Loosely coupled

Documented

Tested

Portable



# Easy

Clean API

Readable code

(minimal macros, no complex macros)

Examples that show common usage patterns

# Loosely coupled

No global code

Only .c and .h for basic modules

(Sym Ciphers / Hash)

Function pointers and hooking for flexibility

(change IP-stack, change certificate verification behaviour, add entropy sources, etc)

# Documented

At least doxygen documented

Extra documentation / tutorials available

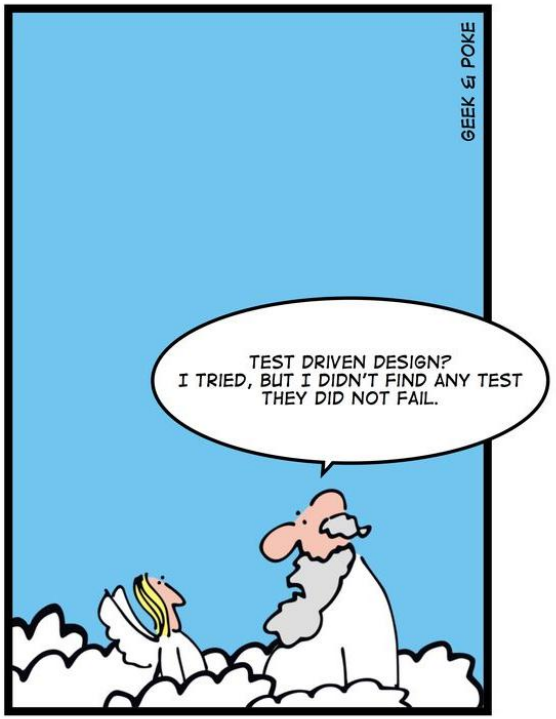
Example code for showing common use cases

# Tested

Auto-generated from basic test code

(Thanks to Ian Blumel for FCTX Unit Testing Framework)

Over 1600 tests for regression, code coverage, cryptographic validation (mostly NIST vectors)



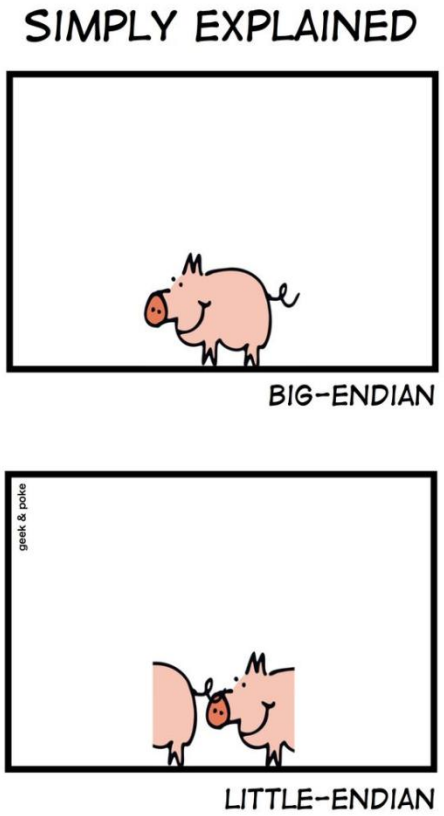
# Portable

ANSI-C (C89 as much as possible)

No global code -> Portability code in module

(Some duplication happens!)

Loosely coupling > DRY



# Simple example

How to make code SSL'ized?

# Headers

```
#include "polarssl/net.h“
#include "polarssl/ssl.h“
#include "polarssl/entropy.h“
#include "polarssl/ctr_drbg.h"
```

# Config

```
ssl_set_endpoint( &ssl, SSL_IS_CLIENT );
ssl_set_authmode( &ssl, SSL_VERIFY_NONE );
ssl_set_rng( &ssl, ctr_drbg_random, &ctr_drbg );
ssl_set_dbg( &ssl, my_debug, stdout );
ssl_set_bio( &ssl, net_recv, &server_fd,
             net_send, &server_fd );
ssl_set_ciphersuites( &ssl, ssl_default_ciphersuites );
ssl_set_own_certificate( &ssl, &crt, &key );
ssl_set_ca_chain( ssl, ca_chain, ca_crl, “mail.google.com” );
```

# Network code

```
while( ( ret = write( server_fd, buf, len ) ) <= 0 )
->
while( ( ret = ssl_write( &ssl, buf, len ) ) <= 0 )

ret = read( server_fd, buf, len );
->
ret = ssl_read( &ssl, buf, len );
```

# Past year

A lot of code improvements

2 Security advisories

OpenVPN-NL accreditation

# Code improvements

Better integration hooks and generic wrappers

PKCS#11 support (through libpkcs11-helper)

PKCS#1 v2.1 (RSAES-OAEP, RSASSA-PSS)

CTR_DRBG (NIST SP 800-90)

Generic entropy accumulator

# Security advisories

**Security Advisory 2011-02 (CVE-2011-4574)**

Weak random number generation within virtualized environments

**Security Advisory 2011-01 (CVE-2011-1923)**

Possible man in the middle in Diffie Hellman key exchange

# OpenVPN-NL accreditation

Dutch landscape

- VIR-BI regulation
- NBV: subsidiary of the AIVD (the Dutch General Intelligence and Security Service) (NLNCSA)

NBV is an evaluation agency that evaluates the protection quality of IT security products and solutions.

# Evaluation

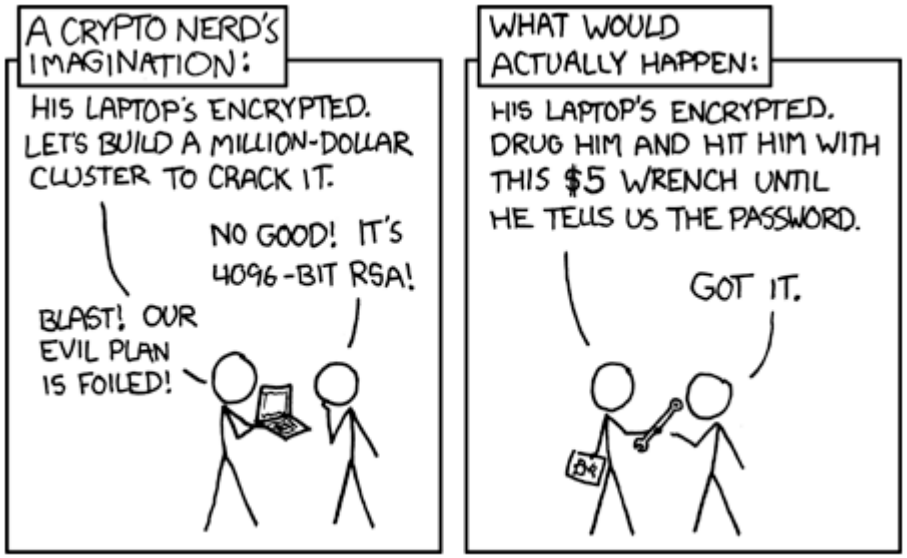
Cryptography

Security features

Security design

Source code inspection

Supply chain control




POLAR SSL

# Stock OpenVPN issues

Insecure configurations possible

- NULL-encryption
- outdated cryptographic functions

Versioning vs development

Supply chain guarantee

Evaluation documentation

# OpenVPN-NL

| | **OpenVPN** | **OpenVPN-NL** |
|---|---|---|
| Maintainer | OpenVPN community | Fox-IT |
| Distribution channel | Various means | https://openvpn.fox-it.com |
| Certification | None | NLNCSA Criteria level 2, “Departementaal VERTROUWELIJK |
| Functionality | Full | Many insecure and less secure options stripped, hardened, otherwise unchanged |
| Crypto-code | OpenSSL | PolarSSL |
| Default | BF-CBC, SHA1 | AES-256-CBC, SHA256 (no other options allowed) |

# Process

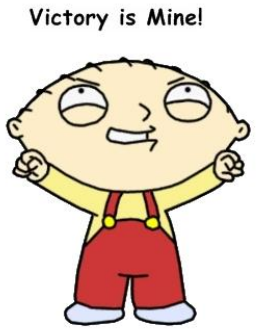
Patches for PolarSSL and OpenVPN

- Hardening
- Integration (Major effort by Adriaan de Jong)
- Source code documentation

Evaluation Documentation

Final versions (last minute changes NOT appreciated ☺)

# In the end




Slashdot

Library Newsletter Jobs

stories
recent
popular

ask slashdot
book reviews
games
idle
yro

technology

cloud
hardware
linux

**Dutch Government Officially Trusts OpenVPN-NL**

Posted by **timothy** on Thursday November 24, @09:02PM
from the goot-to-goeh dept.

First time accepted submitter joost.bijl writes

> "Yesterday the Dutch government took a step to further improve the adoption of Open Source in its ranks. It has officialy approved a modified version of the open source VPN software OpenVPN for use on the governmental level 'Departementaal Vertrouwelijk' (Restricted). The release is called OpenVPN-NL and is fully open-source and available for use. The software has undergone a security evaluation by the Dutch government's national communications security agency (NLNCSA). The major change is the removal of OpenSSL as the cryptographic core of OpenVPN-NL. Instead, the Dutch government opted to include the smaller, better readable and documented open source library PolarSSL to provide the cryptographic and SSL/TLS functionality. The Dutch IT Security company Fox-IT worked together with both OpenVPN and PolarSSL communities and modified the stock software to support the government evaluation process. In total 8000 lines of code and 4000 lines of documentation were checked in to the OpenVPN trunk."

37 of 53 comments loaded | Share

x it x security x yro x encryption x government story

# Contact info

**Paul Bakker**

Email: p.j.bakker@polarssl.org

Twitter: PaulBakkerNL

**PolarSSL**

Site: http://polarssl.org

Twitter: PolarSSL