cryptoColorForth

less is more

Presentation at EuroForth 2017, Bad Vöslau, Austria,

Howerd Oakford www.inventio.co.uk

Aim

To create the simplest possible secure communication, data storage and user interface for the You-Me Drive (YMD).

For more details on the YMD please go to :http://you-me.one/ , a summary is :

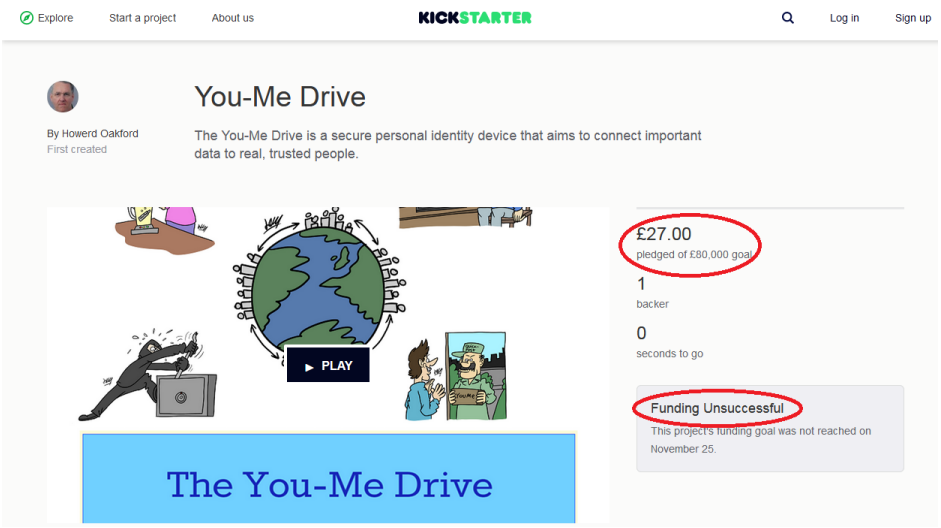
“The You-Me Drive is a secure personal identity device that aims to connect important data to real, trusted people”.

It does this by providing a WiFi USB drive, peer to peer networking software, multi-signed file encryption and a user interface that boots directly on a PC.

Kickstarter

My first, and almost certainly last, Kickstarter project was an attempt to get funding to speed up the development of the You-Me Drive.

https://www.kickstarter.com/projects/inventio/you-me-drive



Even though the Kickstarter campaign was a failure, it provided the impetus to document the project : http://you-me.one/You-Me_Drive_2016Oct01.pdf
I also learned how hard it is to explain a complex idea in a way that appeals to a non-technical audience.

So development continues as time allows, with the current sub-project being the **colorForth** inspired software infrastructure.

Why colorForth?

The YMD requires the highest possible level of security, and this rules out a conventional operating system.

Commercially successful operating systems are designed to lock users into a particular manufacturer's brand of complexity, in order to maximise profitability. The complexity generated to achieve this is so high that it is not possible to evaluate potential security weaknesses.

There are several less complex operating systems, but my personal favourite amongst the FOSS ones has always been **colorForth** - *less is more* in this context.

User-friendly colorForth

There are several versions of **colorForth** published online, Chuck Moore's original (colorforth.com), and many others derived from this, the SourceForge version and the GreenArrays *ArrayForth* GA144 development environment. Since they are all based on Chuck's original version they all share some of Chuck's original design decisions – for example the use of the ANS Forth standard names **or** and **?dup** to have non ANS functionality, the use of 32 bit cell addressing, and the use of the **eax** register for the Top Of Stack instead of **ebx**.

Some of the **colorForth**s have optional QWERTY keyboard text entry, but my view is that this should be available for user text entry only, and not included as part of the programming environment. Also, most **colorForth**s require actual floppy disk hardware to function.

So **cryptoColorForth** (so far) makes some user-friendly changes :

1. ANS Forth names – **or** → **xor** , **?dup** → **qdup**
2. Byte addressing for **@**, **C@**, **!** and **C!**
3. BIOS sector read/write → operation from USB drive
4. QWERTY text input

Further colorForth development

There are many useful functions defined in ANS-like Forths for the x86 architecture - crypto libraries using big numbers, modular exponentiation, TEAN, RSA etc.
With the simple changes listed above, and a meta-data replacement for the conventional file system, **cryptoColorForth** can be made to work with this source code, allowing these building blocks to be imported easily.

Summary

colorForth has an extraordinary "power to weight ratio", its small size (12K bytes for the kernel) and extreme simplicity makes it the ideal platform to develop secure applications, with the You-Me Drive being one example.

Howerd Oakford, 31st August 2017