Matt,

This is a very good summary of the advantages of XTS over the currently approved FIPS 140-3 encryption modes. However, there are a few points, which can be made more accurate:

- Parallel design:
CBC mode is customarily used for large data sets operating on interleaved, separate chains of blocks. There are many applications working on 2..8 such chains, providing sufficient parallelism in storage. Because storage devices are orders of magnitude slower than encryption HW, parallel running encryption chains can still be long. XTS does not provide any advantage over CBC mode for today's storage applications, *in this regard*.

Large block encryption is generally superior for storage (better exploiting data redundancy for authentication). For example, EME2, a large block mode, is also significantly parallelizable, which makes it unclear, if XTS has a parallelizability advantage in practical situations.

- Industry adoption of XTS-AES:
This is not a good argument. Whatever encryption mode gets standardized, it will be adopted by the industry, because customers demand it, manufacturers can save security evaluation, design time, etc. XTS is an IEEE standard encryption mode, for EXTERNAL ENCRYPTION MODULES. Many current implementations use XTS in other settings, for other purposes: wasting resources; leaving security enhancing options unutilized; or even compromising security. Widespread improper use of an encryption mode constitutes arguments against standardization, not for.

- CPU crypto support:
CPU crypto support is mostly sequential. Software-based solutions cannot fully utilize the inherent parallelizability of XTS, other encryption modes are equally accelerated, so XTS has no advantage here.

- Security of XTS
There is a serious concern here: the restrictions on the number of encryption operations. The P1619 standard provides good security bounds while: "the same key is not used to encrypt much more than a terabyte of data". This is not the capacity of the storage device, but the number of encryption operations. In many applications this limit is reached in just hours, requiring partitioning the data into thousands of small bands (using different keys) or daily re-encrypting the stored data with new keys. Either option is very expensive and slows down the system.

- Optional Changes to XTS for NIST Adoption:
These are very useful options.

--- Allow using one AES key for both AES block ciphers: in embedded application secure key storage is expensive.
--- Allow switching the order of the blocks created through Ciphertext Stealing: in disk drives the data is transferred serially at a speed of several gigabits per second. The P1619 standard changed the original ciphertext stealing proposal, and swaps the last two ciphertext blocks. This requires either complicated addressing modes in internal buffers or an extra buffering layer. Either of these introduces latency, requires extra power and gates, design and test efforts, etc. for no apparent reasons.

Laszlo Hars