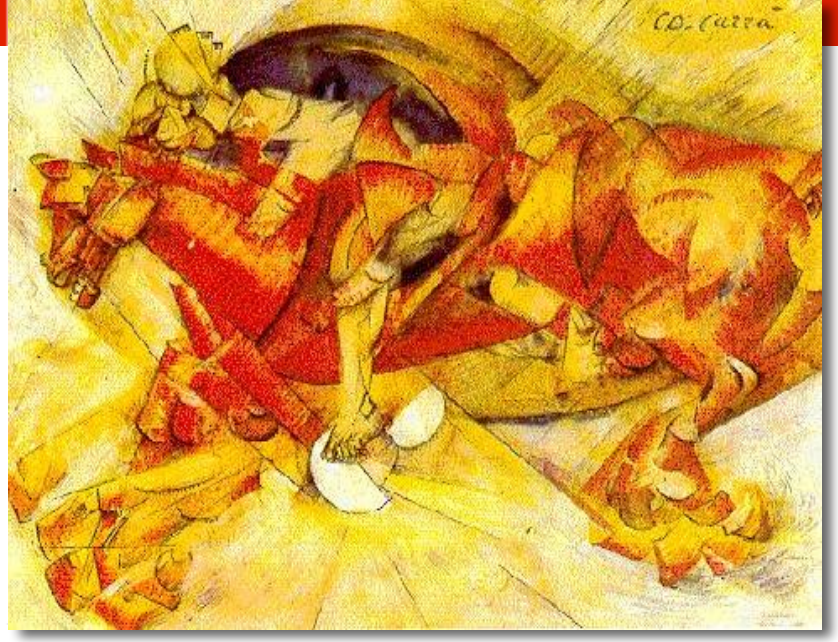
the information security provider

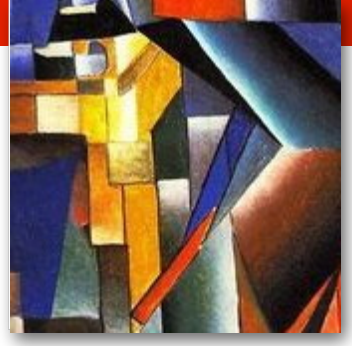


Standing with Integrity
Integrity Testing by way of Sampling

Renaudt Nuñez

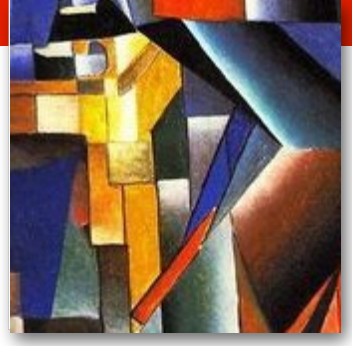
atsec information security corporation

Overview



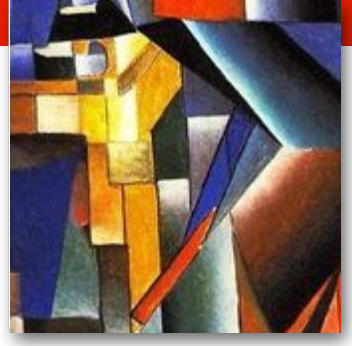
- ✓ **Explain the FIPS 140-2 requirements on Integrity Testing.**
- ✓ **Express the problem vendors face meeting the requirements.**
- ✓ **Give examples of sampling methods performed by other industries.**
- ✓ **Deliver the proposal introduced by Integrity Testing by Sampling working group.**
- ✓ **Question and Answer**

Integrity Testing Defined in FIPS 140-2



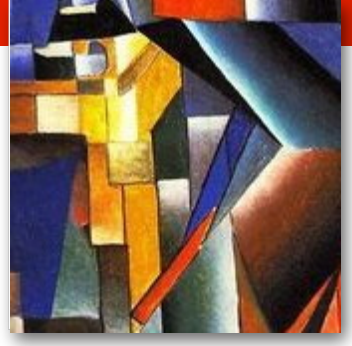
- **The FIPS 140-2 standard requires that…**
  - *The cryptographic module shall perform power up tests: cryptographic algorithm test, software/firmware **integrity test**…* AS09.13 (emphasis mine)
  - *Software/Firmware **integrity test** using an error detection code (EDC) or Approved authentication technique shall be applied to **all** validated software and firmware components…*AS09.22 (emphasis mine)
- **AS.06.08 requires that**
  - *A cryptographic mechanism using an Approved integrity technique shall be applied to all cryptographic software and firmware components within the cryptographic module.*

Firmware Cryptographic Module



- **According to IG 1.3 a firmware module is defined as…**
  - *the programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) within the cryptographic boundary and cannot be dynamically written or modified during execution.*
- **Also…**
  - *If the operational environment is a limited operational environment, the operating system requirements in Section 4.6.1 do not apply.*
- **Meaning that in order to satisfy AS09.22, a firmware module can use an Error Detection Code (EDC) for the integrity test.**

Vendors Current Outlook

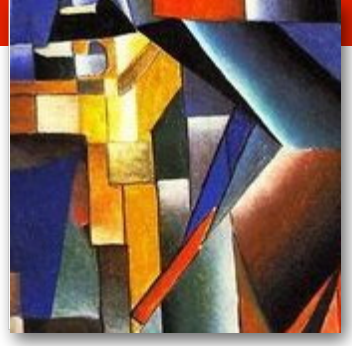


- **With the definitions provided vendors still have two specific problems.**

1) Some modules running on a limited operating environment include large binary images sometimes ranging in the Gigabytes ranges. These large binaries typically only include a small portion of crypto-related content.

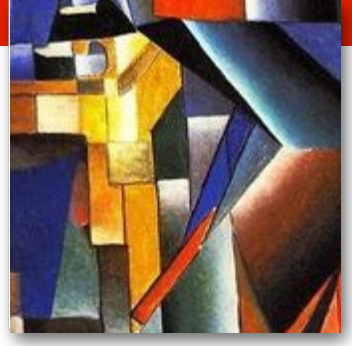
2) Other devices such as smart cards have limited resources. In that case, integrity testing the firmware significantly slows down the power-up process rendering FIPS validated modules inefficient, and sometimes even useless.

What do other industries do?



- **SAMPLING!!**
  - In manufacturing industries, *sampling* is used in quality control; for example, from the number of defects in a sample of electrical components, one can infer the overall reliability of the manufacturing process.
  - In medical research, *sampling* is used to identify health risks; for example, from the difference in heart disease rates between a sample of smokers and a sample of non-smokers, one can infer the overall effect of smoking on the risk of developing heart disease.
  - The most common example people can relate to is that of pollsters. In opinion polls not everyone within a population is interviewed but rather a *sample* is used to draw conclusions as a whole.

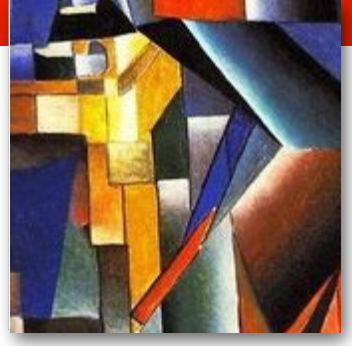
Integrity Testing by Sampling Working Group Saves the Day



- Shortly after a presentation by NIST's own Dr. Allen Roginsky at last year's ICMC, a working group was formed to take on the task of defining a set of rules that could be followed in order to satisfy both worlds.
- This group involved product vendors, CST labs, and participation by CMVP validators.

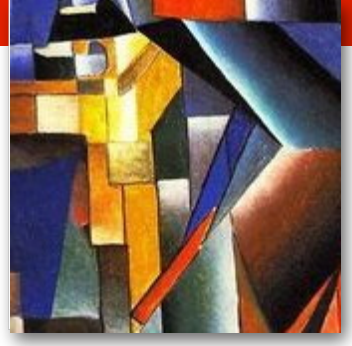
Dr. Allen Roginsky – NIST	Dr. Yi Mao – atsec information security
Renaudt Nuñez - atsec information security	Andrey Jivsov – F5 Networks
Richard Wang - Gossamer	Eric Peeters – Texas Instruments
Shawn Geddis - Apple	Michael Scott - RSA

Integrity Testing by Sampling Outline



- **Assumptions**
- **Initialization of module**
- **Deterministic Sampling**
- **Random Sampling**
- **Error State and Recovery**

Assumptions

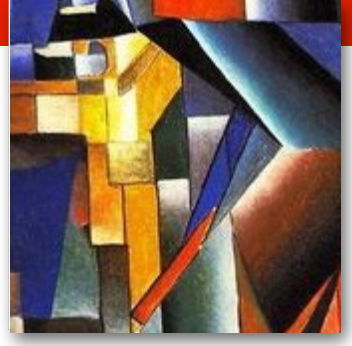


1) A firmware integrity test is a health test only. It is not designed nor is it intended to guard against targeted attacks. The cryptographic module should have other means of defense–commensurate with the module's Security Level–to protect against deliberate attacks.

2) The firmware image that is subject to an integrity test can be subdivided into two parts: (A) crypto-related and (B) non-crypto-related. While it is not possible to exclude (B) from the validation requirements, this partition may be subjected to a less rigorous firmware integrity self-test than (A).

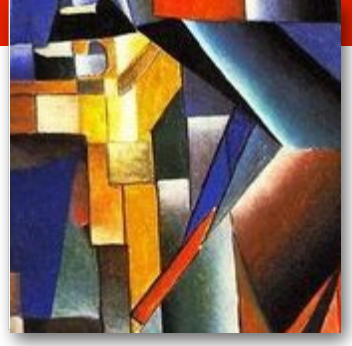
3) The integrity test shall be performed on the entire firmware image during the installation phase. Then we show how each firmware integrity test at each subsequent power-on can be performed on a subset of the entire firmware image.

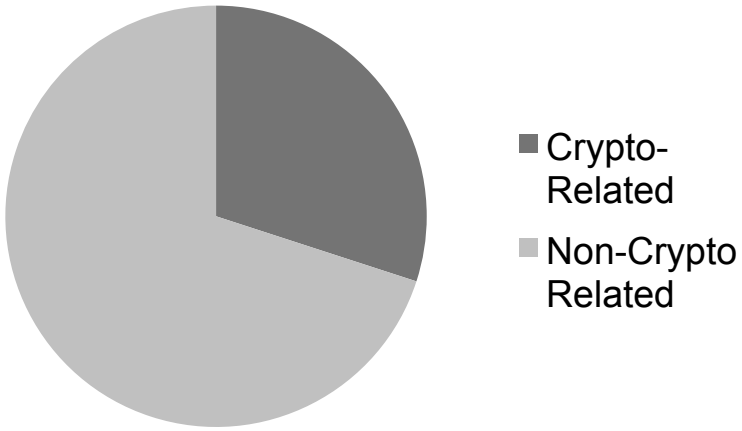
Initialization of the Module

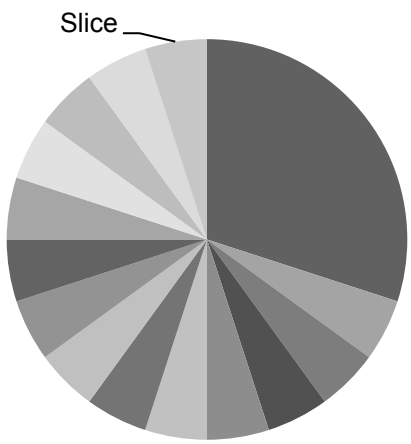


- The crypto and non-crypto parts of the module's firmware are to be tested differently. The entire crypto part **must** be tested every time the integrity test is executed. As for non-crypto portion, a different subset or "slice" of files is selected each time the integrity test is performed.
- If the integrity test on crypto part and on the slice selected from the non-crypto portion are successful, then the module satisfies the firmware integrity test requirement of FIPS 140-2.
- First, the vendor permanently divides the non-crypto files into portions of equal size.
- Next, the vendor chooses to test a slice of the non-crypto files by either **random** selection or by **deterministically** selecting and testing each "slice" sequentially each time the integrity test is performed.

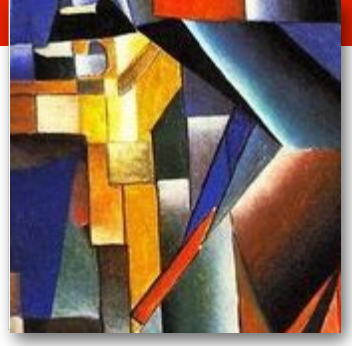
Piece of the Pie





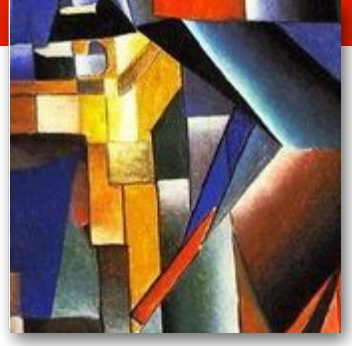


Selecting Segments Deterministically



- **In the most simplistic way, the module may perform an integrity test on the files in a deterministic way.**
- When the deterministic integrity test is performed for the first time, it is performed in the first set. The module saves the set's index counter = 1. Each time the integrity test is performed, the index counter is incremented by 1 and the subsequent files in the next slice are tested. After the last index is tested to be incremented, the counter is set to 1 and the process restarts.

Selecting Segments Randomly

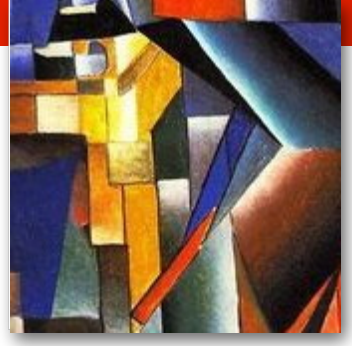


- **The module may use a random number generator to generate integers representing a "slice" of files in the non-crypto related portion.**
  - When a firmware integrity test is performed, the random number generator is called to select a set in the range of the entire non-crypto related portion and then perform an integrity test using EDCs at least 16 bits in length.
  - **Note**: The random number generator used does not have to be approved for use in cryptographic applications; however, it shall generate integers in such a fashion that each "slice" has an equal opportunity to be chosen from the whole.

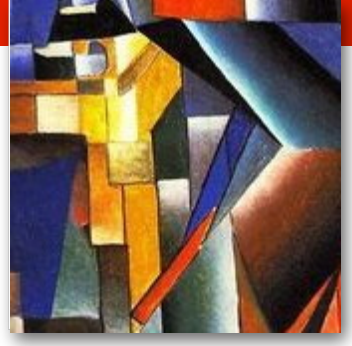
What if I fail?

- **In the event of an integrity test failure, recovery from the error state shall require that the module perform an integrity check of the entire firmware, i.e. crypto and non-crypto related, as was done in the initial installation of the module. If the integrity check of the entire firmware is successful, the module may return to implementing an integrity check using either the random or the deterministic method.**

Final Points



- **Next on the agenda for the draft IG**
  - submitted for review by the CMVP.
  - Draft IG will be sent to the labs for additional input.
- **Vendor/Lab interaction**
  - Voicing your suggestions to the labs as the to benefit of bringing the IG to fruition
  - Defining crypto vs. non-crypto related files



Questions?

-Thank You!