When generating a product key, one of the biggest challenges is to ensure that the key was generated by the product publisher and not by an unwanted party like a pirated key generator. In offline scenarios (when the key cannot be validated over the Internet against a database of generated keys) the key must somehow carry within itself the proof that it is “authentic”. An approach is to layout the key binary data like this: KEY = KEY\_DATA | ENCRYPTED(HASH(KEY\_DATA)) In order to validate the key, the key data is hashed, and the encrypted hash is decrypted. If the computed data hash is identical to the decrypted data hash, it means that the key used to decrypt the data is the correct decryption key. Now this is where it starts to get interesting.

Src : <http://www.softactivate.com/GenerateLicenseKeys.aspx>

* **Generate** a key from only the Motherboard, Processor and BIOS since the user normally doesn't change these parts.
* Don't use MAC ID, Graphics Card ID AND Disk ID since it's very common to change these devices.

Src : <http://www.codeproject.com/Articles/28678/Generating-Unique-Key-Finger-Print-for-a-Computer>

One-way hash

**SoftActivate Licensing SDK Screenshots**