CRYPTO-COMPRESSION OF 3D OBJETS

CONCEPT

KEY PAPERS

Michael Deering, “Geometry Compression”, sun Microsystems, 1995

Jarek Rossignac, “Edgebreaker: Connectivity compression for triangle meshes”, Georgia Institute of Technology, 1999

Jarek Rossignac, “3D mesh compression”, College of Computing and GVU Center Georgia institute of Technology, January 2003

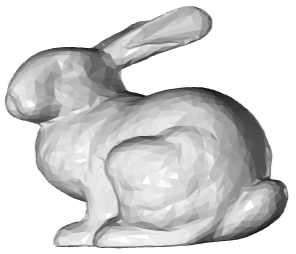
Marc Éluard, Yves Maetz, and Gwenaël Doërr, "Geometry-preserving Encryption for 3D Meshes", Technicolor R&D France, November 2013

First we load a 3D model.

Its vertices, positions and normals are represented by 3 floats (96 bits).

Each triangle is represented by 3 ints, 96 bits

3D MODEL



We quantize vertices positions on a discrete grid, and normals on a precalculated dictionary.

This step compresses data as positions are now represented on 30 bits and normals on 17 bits.

QUANTIZATION

Edgebreaker compresses the mesh by representing triangle strips on a low entropy format.

This step can squish a triangle in 1.5 bits, but 3 bits is a more realistic representation for worst case scenarios.

EDGEBREAKER

Then, we use a Geometry Preserving Encryption.

This is an encryption algorithm capable of preserving the 3D nature of our data while concealing its shape.

ENCRYPTION

Finally, an entropic compression can be done to further increase the compression ratio.

ENTROPIC COMPRESSION

ENTROPIC DECOMPRESSION

First, we decompress our file.

DECRYPTION

Then we decrypt our data using a key. As the algorithm is geometry preserving, it will give us inteligible data even without the right key.

EDGEBREAKER

We use the edgebreaker decompression to reconstruct our mesh.

3D MODEL

Finally, a 3D model is returned.

That 3D model could either be "scrambled" if we used the wrong key during decryption, or a quantized mesh if we used the right key