COLLOID LLC

UBERCRYPT FRAMEWORK

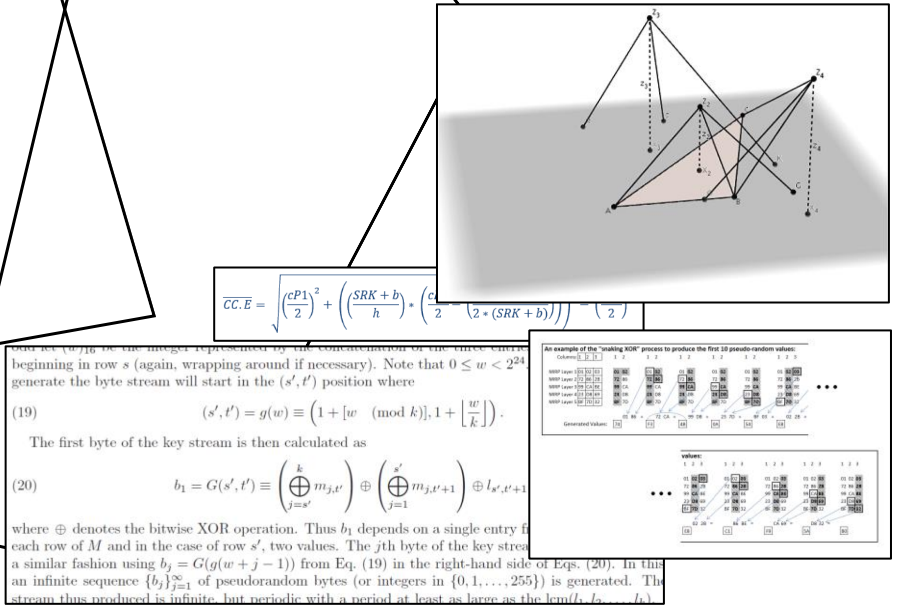
A Lean Briefing

ABOUT THE UCF

www.UberCrypt.com

The UberCrypt Framework is more than just a symmetric key stream cipher, it is a cryptographic **framework** with unique and dynamic properties. The benefits of which result in an unparalleled combination of security, speed and flexibility. It is an "**enabling technology for data security**."

(US Patents: 8767954 & 9118481)



$\overline{CC.E} = \sqrt{\left(\frac{cP1}{2}\right)^2 + \left(\left(\frac{SRK+b}{h}\right) * \left(\frac{c}{2} - \ldots (2*(SRK+b))\right)\right) - \left(\frac{\ldots}{2}\right)}$

beginning in row s (again, wrapping around if necessary). Note that $0 \le w < 2^{24}$ generate the byte stream will start in the (s', t') position where

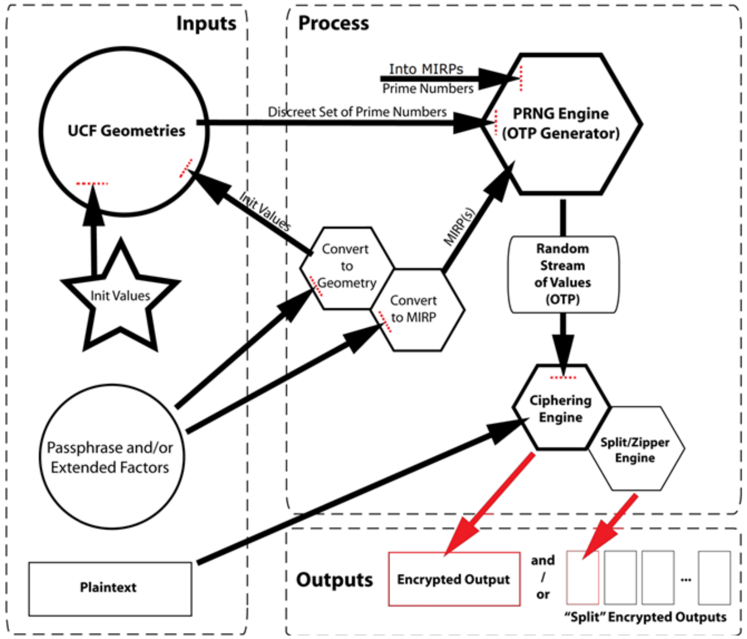
$$(s',t') = g(w) \equiv \left(1 + [w \pmod k], 1 + \left\lfloor \frac{w}{k} \right\rfloor\right). \tag{19}$$

The first byte of the key stream is then calculated as

$$b_1 = G(s',t') \equiv \left(\bigoplus_{j=s'}^{k} m_{j,t'}\right) \oplus \left(\bigoplus_{j=1}^{s'} m_{j,t'+1}\right) \oplus l_{s',t'+1} \tag{20}$$

where $\oplus$ denotes the bitwise XOR operation. Thus b_1 depends on a single entry f each row of M and in the case of row s', two values. The jth byte of the key strea a similar fashion using $b_j = G(g(w+j-1))$ from Eq. (19) in the right-hand side of Eqs. (20). In this an infinite sequence $\{b_j\}_{j=1}^{\infty}$ of pseudorandom bytes (or integers in $\{0, 1, \ldots, 255\}$) is generated. The stream thus produced is infinite, but periodic with a period at least as large as the $\mathrm{lcm}(l_1, l_2, \ldots, l_k)$.

The UberCrypt Framework



THE UBERCRYPT *FRAMEWORK*

PROPERTIES

WHAT MAKES IT UNIQUE?

Symmetric Key Stream Cipher:
Encrypt/Decrypt data at speeds in excess of **3Gbps in software**, even faster on silicon. Supports **bit-granular encryption key strengths** ranging from ~250 to ~3000 bits (or more). Speed is **independent of key strength**. Multiple **distributed key architecture**. Built on 3D geometry, allows for a **virtual infinity of geometries**.

Math at: International Assoc of Cryptologic Research (IACR): http://eprint.iacr.org/2014/894

BENEFITS

HOW CAN IT BENEFIT YOU?

Security, Speed, Flexibility, Customization:
Elastic key strength makes brute force attack on keys orders of magnitude harder than something like AES. Same speed at any strength amplifies this benefit. Distributed, user/admin selected keys makes theft attacks virtually impossible. Infinite geometries makes "*mass customization*" of data encryption possible – imagine every person or device having their own cryptosystem.

APPLICATIONS

WHERE CAN IT BE USEFUL?

Secure communications (particularly video-conferencing). Streaming media. Digital Rights Management. Very Large Databases/Files (byte level decryption). Crypto currency/blockchain. Random number generation for simulations and other applications. Many more…

POSSIBILITIES

WHAT MIGHT THE FUTURE HOLD?

Future-Proof Security is Quantum Resistant. **Possibly the first symmetric and asymmetric key system**. Digital signing. Three-Party Encryption. More…

REAL, NOT IMAGINARY

NIST STS PROVEN

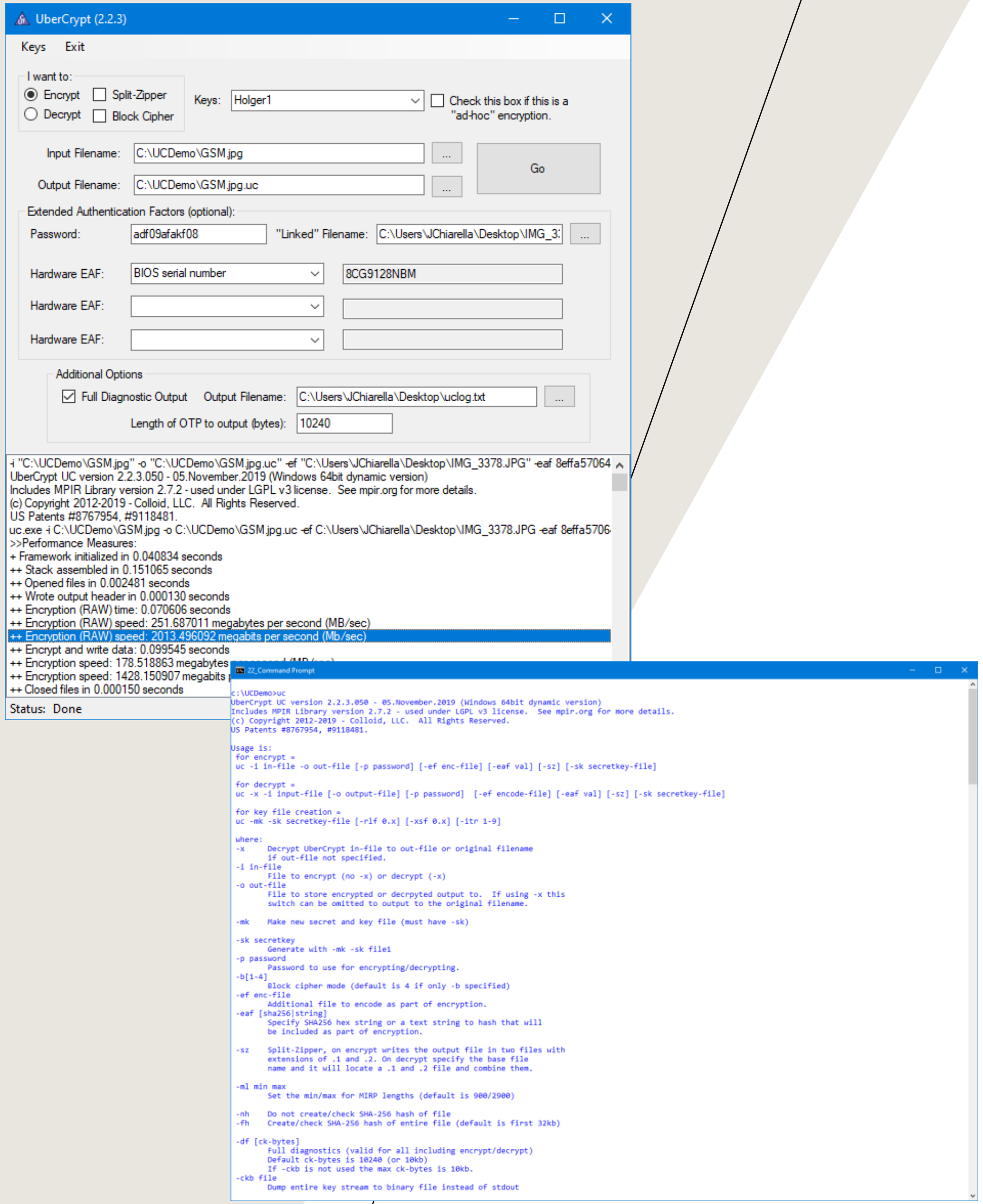
Billions of tests using NIST's Statistical Testing Suite (STS) measuring quality of cipher key stream. Scored higher on random data generation than the quantum photonic generator at the Australian National University.

SOFTWARE

Runs on Windows, Linux, OSX. (Native C/C++)
Will port to silicon and mobile.
Stable, extensively tested, zero failures.

API-ENABLED

Easily integrated into other applications via API



I want to:
● Encrypt □ Split-Zipper
○ Decrypt □ Block Cipher

Keys: Holger1

□ Check this box if this is a "ad-hoc" encryption.

Input Filename: C:\UCDemo\GSM.jpg

Output Filename: C:\UCDemo\GSM.jpg.uc

Extended Authentication Factors (optional):

Password: adf09afakf08 "Linked" Filename: C:\Users\JChiarella\Desktop\IMG_3:

Hardware EAF: BIOS serial number 8CG9128NBM

Hardware EAF:

Hardware EAF:

Additional Options

☑ Full Diagnostic Output Output Filename: C:\Users\JChiarella\Desktop\uclog.txt

Length of OTP to output (bytes): 10240

```
-i "C:\UCDemo\GSM.jpg" -o "C:\UCDemo\GSM.jpg.uc" -ef "C:\Users\JChiarella\Desktop\IMG_3378.JPG" -eaf 8effa57064
UberCrypt UC version 2.2.3.050 - 05.November.2019 (Windows 64bit dynamic version)
Includes MPIR Library version 2.7.2 - used under LGPL v3 license.  See mpir.org for more details.
(c) Copyright 2012-2019 - Colloid, LLC.  All Rights Reserved.
US Patents #8767954, #9118481.
uc.exe -i C:\UCDemo\GSM.jpg -o C:\UCDemo\GSM.jpg.uc -ef C:\Users\JChiarella\Desktop\IMG_3378.JPG -eaf 8effa5706
>>Performance Measures:
+ Framework initialized in 0.040834 seconds
++ Stack assembled in 0.151065 seconds
++ Opened files in 0.002481 seconds
++ Wrote output header in 0.000130 seconds
++ Encryption (RAW) time: 0.070606 seconds
++ Encryption (RAW) speed: 251.687011 megabytes per second (MB/sec)
++ Encryption (RAW) speed: 2013.496092 megabits per second (Mb/sec)
++ Encrypt and write data: 0.099545 seconds
++ Encryption speed: 178.518863 megabytes per second (MB/sec)
++ Encryption speed: 1428.150907 megabits p
++ Closed files in 0.000150 seconds
```

Status: Done

22_Command Prompt

```
c:\UCDemo>uc
UberCrypt UC version 2.2.3.050 - 05.November.2019 (Windows 64bit dynamic version)
Includes MPIR Library version 2.7.2 - used under LGPL v3 license.  See mpir.org for more details.
(c) Copyright 2012-2019 - Colloid, LLC.  All Rights Reserved.
US Patents #8767954, #9118481.

Usage is:
 for encrypt =
 uc -i in-file -o out-file [-p password] [-ef enc-file] [-eaf val] [-sz] [-sk secretkey-file]

 for decrypt =
 uc -x -i input-file [-o output-file] [-p password]  [-ef encode-file] [-eaf val] [-sz] [-sk secretkey-file]

 for key file creation =
 uc -mk -sk secretkey-file [-rlf 0.x] [-xsf 0.x] [-itr 1-9]

 where:
 -x     Decrypt UberCrypt in-file to out-file or original filename
        if out-file not specified.
 -i in-file
        File to encrypt (no -x) or decrypt (-x)
 -o out-file
        File to store encrypted or decrpyted output to.  If using -x this
        switch can be omitted to output to the original filename.

 -mk    Make new secret and key file (must have -sk)

 -sk secretkey
        Generate with -mk -sk file1
 -p password
        Password to use for encrypting/decrypting.
 -b[1-4]
        Block cipher mode (default is 4 if only -b specified)
 -ef enc-file
        Additional file to encode as part of encryption.
 -eaf [sha256|string]
        Specify SHA256 hex string or a text string to hash that will
        be included as part of encryption.

 -sz    Split-Zipper, on encrypt writes the output file in two files with
        extensions of .1 and .2. On decrypt specify the base file
        name and it will locate a .1 and .2 file and combine them.

 -ml min max
        Set the min/max for MIRP lengths (default is 900/2900)

 -nh    Do not create/check SHA-256 hash of file
 -fh    Create/check SHA-256 hash of entire file (default is first 32kb)

 -df [ck-bytes]
        Full diagnostics (valid for all including encrypt/decrypt)
        Default ck-bytes is 10240 (or 10kb)
        If -ckb is not used the max ck-bytes is 10kb.
 -ckb file
        Dump entire key stream to binary file instead of stdout
```

MEET THE TEAM

Joe Chiarella

Serial Tech Entrepreneur

Seven companies
Two 8-digit exits on two cyber-security software companies
(PP to CA and XPL to AVG)

Strong experience with business, product strategy & management. Experience with “big data” and analytics. Experienced with M&A.

www.Linkedin.com/in/josephchiarella

www.JoeChiarella.com

Greg Mosher

Software Engineering Executive

Five companies
Two 8-digit exits on two cyber-security software companies
(PP to CA and XPL to AVG)

Led large engineering organization of 200+ distributed globally. Experienced with M&A.

www.Linkedin.com/in/gregamosher

COLLOID LLC

UBERCRYPT FRAMEWORK

SUMMARY

www.UberCrypt.com

The **UberCrypt** ***Framework*** is a data security enabling technology. It is particularly well-suited to high-volume, high-speed, low-latency data security demands ranging from commercial to national defense and even orbital platforms. Unique properties make it dynamic and customizable.

Patents and software are available for (exclusive or non-exclusive) license or purchase.

Contact:

Joe Chiarella