L.I.T.E.
Limited Information
Tetration Encryption

Darcy olvelieu

Ben puzzleduck

What's wrong with encryption?

- Based on computational complexity
- Everything encrypted today will one day be decryptable
- The richest can now buy decryption on demand
- SHA1 has fallen and SHA256 is in doubt
 - Quantum computing is the answer we have explore

Can we fix it?

- An encryption scheme without dependency on computational power
- Dependant on fundamental laws of nature for security assurance
- Drop in replacement for RSA and SHA
- For reference there are 1080 atoms in the universe
- Tetration creates numbers in the order of 10^{15,151,335} using single digit inputs

Tetration What is it?

Incrementing ++

$$-((3++)++)++=6$$

Addition + (Repeated increment)

$$-3+3 = ((3++)++)++ = 6$$

Multiplication * (Repeated adds)

$$-3*3 = 3+3+3 = 9$$

 Exponentiation 1 / Power of ^ (Repeated multiply)

$$-3*3*3 = 3^3 = 3^3 = 3^3 = 27$$

 Tetration / superexponentiation / hyperpower 11 (Repeated exponent)

$$-31(313) = 3113 = 3^3...(27 \text{ times})...^3$$

• (result is > 5gb)

How does it work?

- Just like private key cryptography
- Private key: eg 581147 Not capable of being stored in this universe
- Public key: subset of digits from Private key and a subset range of the result
- Encrypted data can utilise 100% of a channel to transmit a message of unknown length or values (Perfect forward security)
- Second order recursion (slow with standard computation)

How does it work?

Private Key (b11p): 581147

870270548720859790408294679847698427498074390875298757809285 7938275 932487297353790253975927394572985723897698638 92634587 542985798743059278394769447427469876439474296933273934759074 3537593279457487297529759273497854927905729304750297594023....

Public Key Range: (tl, th)

Public Slice Value (g): 932487297353790253975927394572985723897698638

$$C = trunc_both(M \uparrow \uparrow g, t_l, t_h)$$

$$M = {^{trunc_both(b\uparrow\uparrow p,t_l,t_h)}}\sqrt{C}$$

Further Details

- Example C code in GitHub repository
- Worked example in text
- Exploration of concepts
- Presentation
- https://github.com/ivelieu/LITE
- https://ivelieu.github.io/LITE/

Questions? Comments? Suggestions?

Darcy olvelieu

Ben puzzleduck