Computational Function Transformation (CFT)

Insanely Better Encryption by CFT

Peter Lablans
CEO/CTO at LabCyfer
Published Apr 1, 2025

Threats of quantum computing and 'Harvest Now, Decrypt Later' (HNDL) attacks demand a fundamental reassessment of encryption strategies. Increased Advanced Persistent Attacks (**APTs**) have attackers reside in Cloud Servers, observing and analyzing cybersecurity practices for attacks. Standard cryptographic algorithms like AES-GCM and ChaCha20, while currently robust, face future compromise. This should make exclusive reliance on current encryption of ciphertext being unbroken for the next decade at least somewhat risky.

Shifting Away from the Limiting XOR

The most widely modern applied encryption methods are AES-GCM and ChaCha20. While modern in the sense of computerized “confusion” and “diffusion” it is at least 70 years old in its practice of combining a plaintext with a keystream through a simple XOR operation. And strangely, is almost identical to the old WW2 Lorenz Cipher.

The exclusive reliance on the Exclusive Or (XOR) in encryption is an extremely limiting approach that relies entirely on the security of the keystream.

The Computational Function Transformation (CFT) method, provides a dramatic change in number of possible combining functions, using simple computational transformations, replacing the XOR. It may seem that XORs are the only useful computational combining function. But they are not. The space of (reversible) combining functions is insanely large, exceeding the solution space of secret keys by incredibly large powers of magnitude and reaching in variations of the order of 10^500 and larger. (and this is not a typo).

Computational Function Transformation (CFT)

Computational Function Transformation (CFT) is a theory based approach to encryption that is inspired by the architectural/implementation computer design framework developed by **Dr. Gerrit "Gerry" Blaauw**, my professor in computer design. Dr. Blaauw, a computing pioneer, was one of the key architects on the legendary IBM System/360. Blaauw distinguishes between architecture, implementation and physical realization of a system.

CFT is a transformation of computational implementation, while preserving its architecture. One CFT transformation is the patented **Finite Lab-Transform or FLT**. The FLT applies n-state reversible inverters, basically a series of n unique n-state elements. There are factorial of n (n!) different reversible n-state inverters. For a byte-oriented operation (n=256) that means more than 10^500 variations. The security of the CFT solution space is achieved by its astronomical unpredictability, and not by keeping the type of transformation secret.

The CFT Solution Space

The solution space in the **Advanced Encryption Standard (AES)** and **ChaCha20**, is defined by its secret key. AES usually has a secret key of 256 bits with 1 in 10^77 security. For convenience, the computational security strength of the steps of AES is often considered to be infinite. Under real-life attacks its security is much smaller.

The CFT solution space is formed by combinatorial explosion effect of a transformation, as illustrated in the FLT.

Another function transformation is a use of random 256-state 2 operand operations. For n=256 that is a number 256^(256^2) or greater than 10^150,000. (insanely large).

The solution space of CFT therefor, as in AES for instance, is much larger than its key-space. This creates a new and unheard level of security.

Integration of CFT Customization in Public Key Infrastructure (PKI)

CFT seamlessly integrates with existing PKI frameworks, leveraging quantum-resistant key generation (e.g. Kyber). A secret 32-byte key can generate billions of different reversible inverters. Crucially, CFT can be implemented with minimal modifications to existing Key Management Systems (KMS).

Conclusion

Computational Function Transformation (CFT) applies the preservation of a system architecture while modifying its logic implementations. It has an incredibly large solution space, well beyond the security provided by a secret key. It's not just 'stronger'—it creates an entirely different standard.

Try It/Test It

With CFT we're talking about security levels that are almost ridiculous, in a good way. The Blaauw approach in architecture preservation ensures the integrity of operation. The CFT creates immense and unpredictable variations. Working implementations of CFT in Python, C, and Matlab are available for **free download and for trial and testing purposes only, here**.

P.S.

As pointed out by some: I am aware of the unlikely astronomical space size. And yes, I am also somewhat puzzled why other, better known, cryptographers have not applied the concept. But they haven’t and I have. And it is novel and non-obvious.