# Technical Report HCIA

by Protego Information AB

*By Bo Dömstedt, MSc Chief Cryptographer & Mats Stenfeldt, Director of Development*

*Introduction*

The encryption and decryption of information has long been an important tool for preventing unauthorized and undesired access to secret information, whether this information is stored in a computer, on a computer-readable storage medium, or transmitted between two parties over some communication link. With the evolution of computers and telecommunications technology, the quantity of information created and exchanged on a day to day basis is ever increasing and ever more accessible. The need to prevent unwanted access to, and possible tampering with, this information in a manner that is rapid to implement but ensures high security is therefore greater than ever before.

*Conventional Symmetric Cryptography*

Encryption and decryption schemes typically rely on the use of an algorithm in combination with a data sequence or so-called cipher key. Conventionally, symmetric algorithms, wherein the sender and receiver (or creator and reader) of information share the same secret key, are most widespread. These schemes generally fall into one of two classes, stream ciphers and block ciphers. An example of the latter scheme is the Data Encryption Standard (DES). In such a scheme, the algorithm is time-invariant. In other words two different messages (plaintext) encrypted with the same key will undergo an identical series of computational steps. Depending on the algorithm, a change of key may alter the computation only slightly. Symmetric algorithms as they exist today are basically software or hardware implementations of electromechanical devices invented before the computer. They are not really suitable for software implementations. It is like simulating a mechanical watch in software or hardware. The key size is typically in the domain of 56 to 256 bits.

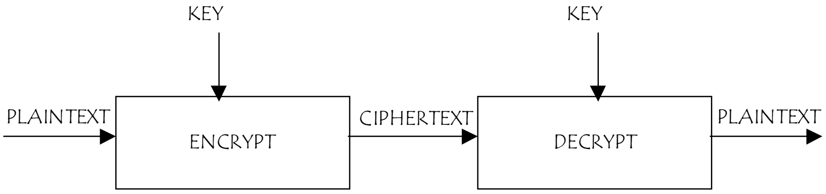

**Fig. 1. General encryption/decryption process for stream ciphers and block ciphers**

*Public-Key Encryption*

Today public-key encryption has gained much popularity for usage in encryption systems. The method that is used in most systems is the RSA variant. The user of the system has two keys, one public and one private. You can use the receiver's public key to encrypt a message, but only the receiver can decrypt the message with his private key. Public-key encryption techniques is based on trap-door one-way functions. Encryption is the "easy" direction into the trap-door. Decryption without the private key is the "hard".

This method of encryption is very slow. Practically it's only usage is to exchange keys for symmetric cryptography. The key size is typically in the domain of 512 to 1024 bits. Like almost all other ciphers, the security provided by public key systems is dependent upon the computational complexity of breaking them. Since these new cipher systems are based on rather simple and elegant mathematical properties, new mathematical knowledge could weaken them much more quickly than in the case of cipher systems that are based on more arbitrary complexity.

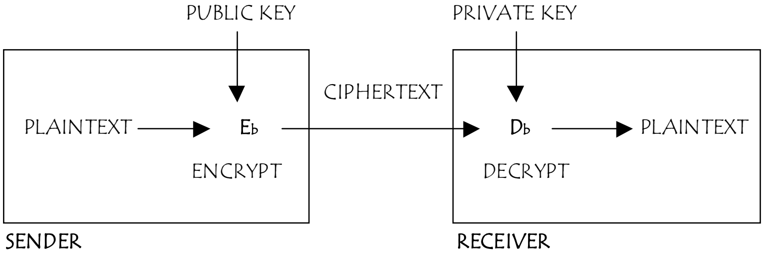

**Fig. 2. General encryption/decryption process for Public-Key Encryption**

*HCIA - a New Way of Encryption*
With the introduction of our encryption method the High Complexity Interpretation Algorithm (HCIA) a new technology for encryption is born. The HCIA system is based on the irreversibility of executional output and description problems of language. This could sound complicated (it's not trivial anyway), but we will give it a try.

The use of static algorithms and time-invariant methods as in conventional symmetric cryptography is avoided by interpretation of the key and the message as an encoding of a new algorithm. Practically you will get a new system or loosely speaking a new algorithm for every encryption you make. HCIA is an "encryption system creator". This gives the cryptanalyst who tries to break the system a "new algorithm", that is unknown to him, to analyze for every intercepted message.

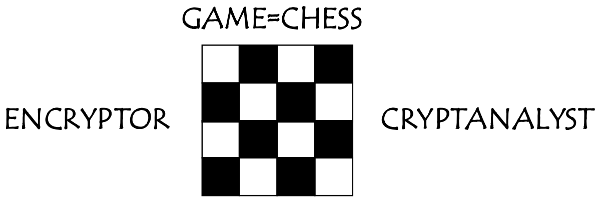

**Fig. 3. Conventional Game**

Breaking a system is like a good game of chess. White is the encryptor and black is the cryptanalyst who tries to break the system. We could compare the game of chess to a known encryption method as for example the Data Encryption Standard (DES). Our method HCIA presents a new problem for the cryptanalyst - he doesn't know what game we are playing with him. The game is not chess anymore it's to determine what game the encryptor is playing with the cryptanalyst.

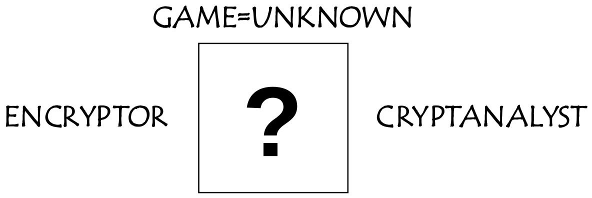

**Fig. 4. The HCIA Game**

*Implementation of HCIA*
The method of encryption for HCIA is execution. The execution is carried out in a form that is well suited for implementation in both software and hardware. By using more memory than conventional ciphers HCIA can execute more efficiently on modern digital equipment.

*Speed of HCIA*
The heart of HCIA is the instruction set. The instruction set could be compared with the instruction set of a microprocessor. But in HCIA the instructions are crypto-operations that

perform cryptographic work. By increasing the number of instructions the security of the HCIA system will be substantially increased. This will not affect the execution time of the cipher. Increasing the number of iterations in any conventional cipher, on the other hand, will make that cipher slower. This is a key point, the strength of the HCIA system may be adjusted without any speed penalty. The HCIA system can easily be pipelined. This is the same method that is used in supercomputers for speeding up calculations. Design estimates for a hardware implementation in ASIC technology gives encryption/decryption speeds up to 6.4 gigabit per second.

*Practical Applications*

The fact that HCIA can retain its security independent of the speed of execution, makes high speed server side encryption possible. Deploying encryption on a Virtual Private Network with 100 connections on the server side with conventional cryptography would kill the server by draining the CPU of too many cycles for the purpose of encryption/ decryption. Video on demand encryption is also possible by utilizing HCIA's efficiency and speed. HCIA can encrypt the video when it is distributed to the viewer with the viewer's individual key in real time thus avoiding having one key for all viewers.

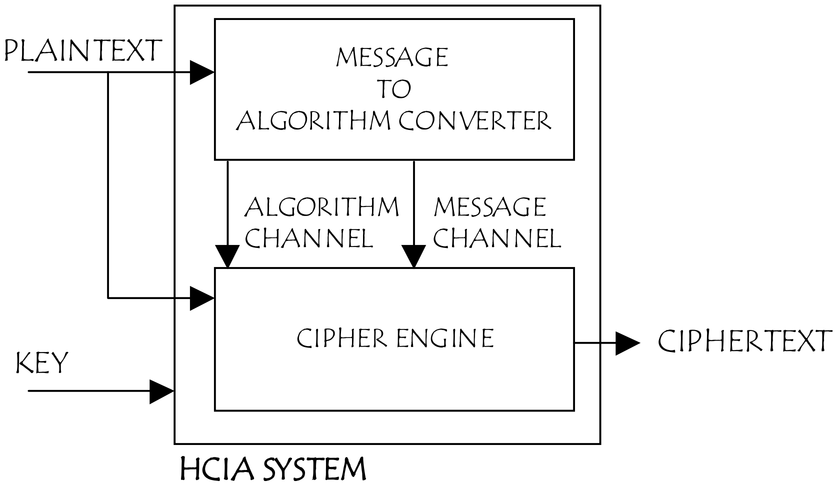

**Fig. 5. HCIA Encryption/Decryption**

*On Uncomputable Functions*

In mathematics it was generally assumed that if an integer function $y=\Omega(x)$ exists, then there must also exist a method $\mathscr{M}$(input,output) for computing that function. But in 1936 Alan Turing, Alonzo Church, and others, discovered that there are more integer functions $\Omega$ than there are computational methods $\mathscr{M}$. Based upon the Diagonal Method by Georg Cantor 1891, we can prove that there are as many integer functions $\Omega$ as there are points on a continuous surface (of infinite size) i.e. uncountable many. Alan Turing found, by applying his Universal Machine that the number of all possible calculations $\mathscr{M}$ are countable, corresponding to the set of integers. The existence of uncomputable functions $\Omega$ follows as there are more functions $\Omega$ than there are calculation methods $\mathscr{M}$; we cannot completely cover a continuous surface with points.

Functions $\Omega$ that lack a corresponding computational method $\mathscr{M}$ are *uncomputable*.

*More Cryptanalysis*

We assume that a cryptanalyst may obtain several ciphertexts $C_0,C_1,...,C_n$ encrypted with the same secret key K. The cryptanalyst may also possess partial or statistical knowledge $m_0,m_1,...,m_n$ of the corresponding plaintexts $M_0,M_1,...,M_n$ Some plaintexts may be completely known by the cryptanalyst, $m_i=M_i$.

As the HCIA system uses a finite secret key K it is obvious that the secret key can be uniquely defined from the information available to the cryptanalyst $\{C_0,C_1,...,C_n,m_0,m_1,...,m_n\}$.

We may express that with a function: $K=\Omega(\{C_0,C_1,...,C_n, m_0,m_1,...,m_n \})$. The cryptanalyst now try to compute the function $\Omega$ by using some systematic computational method $\mathscr{M}$ that reads the input and writes the secret key K: $\mathscr{M}(\{C_0,C_1,...,Cn,m_0,m_1,...,m_n \},K)$.

We now have the remarkable situation that as a direct consequence of that the HCIA cipher is constantly changing the rules of game, we can show that the function $\Omega$ is uncomputable, there is no computational method $\mathscr{M}$, *hence no cryptanalytical attack exists*.

*Rice's Theorem*

The HCIA cipher use the secret cipher key K and the input plaintext M as the input software for a generating machine. A generating machine is a Universal Turing Machine that use its input as a software and generates strings in some language $\mathscr{L}$. We assume that the cryptanalyst has obtained some properties of the output, and try to convert this into a corresponding knowledge of the input.

Due to a result by H. G. Rice 1951, nowadays well known as Rice's Theorem, we know that any nontrivial property of the languages generated by a generating machine is uncomputable. For any specified output property it is not possible to divide the set of inputs into two nonempty halves where only one half satisfies the output property. We conclude that the knowledge that the cryptanalyst has obtained about the output cannot be translated to a corresponding knowledge of the input.

*Other Attacks*

Rice's theorem applies to nontrivial properties of the output. The HCIA system may still be attacked exploiting trivial properties of the system. The most obvious attack would be to trial decrypt every possible key.

## Summary

The High Complexity Interpretation Algorithm is a new way of encryption. The security of the system is based on mathematically provable hard problems. The system creates a new "algorithm" for every message. The execution speed is independent of the security. The HCIA is designed for implementation in modern digital equipment. The HCIA has worldwide patents pending status.

*Legal notes:*