The University of Texas at Austin
Center for Identity

Chaotic Hybrid Encryption Communication Kit

Benito R. Fernandez
Jose R. Capriles Carlos
A. Garcia

2015

UT CID Report #1510

This UT CID research was supported in part by the following organizations:

LifeLock Guarantee Your Good Name · HID · TransUnion · Applied Fundamentals Consulting LLC · CSID · LexisNexis · Safran MorphoTrust USA

Contents

1 Abstract

This research was conducted to design and deploy a new mechanism for data encryption using a combination of a hybrid processor and chaotic oscillators. Our mechanism generates an “infinite key” used as input in standard encryption algorithms. A prototype was developed to show that our patent-pending method works and can be used to reduce the risk of current Technologies.

The first phase involved an extensive review about modern cryptography, chaotic encryption, chaotic oscillators and its integration with a hybrid (mixed-signal) processor. During the second phase, we selected the tools and languages to implement the software prototype. Third, we developed a crypto-tool that includes the following modules: hybrid (mixed-signal) processor simulator, chaotic oscillators, key generator, encryption/decryption tool and GUI (Graphical User Interface). In the final phase, we have done quality assurance and testing of the tool using proprietary scripts and the GUI. The results can be visualized using two instances of the crypto-tool running with different users and two instances of the message consoles. The GUI’s shows the conversation between two parties and the consoles show how the information is encrypted and securely transmitted. A future research will consider multi parties and a hardware prototype.

2 Introduction

2.1 Motivation

Since the *homo erectus* humans have had the right to whisper in someone’s ear, anywhere, anytime. We are losing this right in the modern world. The type and amount of information that people need to transmit securely continually increases everyday. At the same time, the number of spies, hackers and cyber-attacks to public and private entities has been increasing at alarming rates. In fact, there have been many instances where private information has been compromised and available to the public. Clearly, privacy protection is a very important challenge for corporations and the average citizen and demands focus in research.

Today encryption is paramount for *secure* (private) communications. There are several patented methods of encryption available commercially. Most techniques[1, 2] require finite encryption keys or provide fixed-circuit designs.

2.2 Problem Statement

There is no “absolute security”. The term **security** means that *under some given assumptions about the system, no attack of a certain form will compromise the system’s specifications*.

With this in mind, NERDLab (Neuro-Engineering Research & Development Laboratory) has assembled a multi-disciplinary team of engineers to design and deploy a new mechanism for data encryption that reduces the risks of the current technologies.

Based on mixed-signal circuitry patents [3, 4, 5], we have used the hybrid processor as key differentiator of our research. Our solution generates an "infinite key" (∞-key) and is grounded on software-reconfigurable hardware (FPHA–Filed-Programmable Hybrid Array), adding several layers of complexity that may render any attempt to break it futile.

This research demonstrated that our hypotheses was correct. First, the hybrid processor is totally integrable with the chaotic oscillators capabilities to create an ∞-key for encryption. Second, the new method works and could be used in the current Cyber Security Market. Lastly, the obtained results shows a good performance of the encryption method. However, a more exhaustive testing should be executed in futures phases of the on-going research to have a better benchmarking and performance evaluation.

2.3 Key concepts related with the Research

We have decided to introduce this section, with the idea to unify concepts around our research. On the other hand, for those readers who are not familiar with this topic, we are mentioning some of the important concepts used during this study.

Hybrid (mixed-signal) processor: Researchers at The University of Texas at Austin have developed intellectual property that allows designing and building a hybrid (mixed-signal: analog and digital) processor for solving systems of ordinary differential equations. The technology merges analog and digital circuitry collectively to overcome computational difficulties associated with either computer platform. The machine would consist of a multitude of interconnected, interacting, mixed-signal integrator cells, augmented with a mixed-signal nonlinear function approximator. All integrations and calculations are time continuous and over the set of real numbers, using a mixed-signal floating-point representation consisting of exponent, mantissa, and *analog bit*$^{\text{TM}}$. The hybrid system is not a sampled data system, but a true mixed-signal processor. Performance, speed, accuracy, and size will substantially outperform modern supercomputers at a minisscule fraction of the cost, size and power consumption. In addition, the technology permits signal processing without analog-to-digital converters. The new platform could be embodied as a co-processor, giving a laptop or hand held device supercomputer capability.

Chaotic Systems: A chaotic system, is a type of complex non-linear system characterized by sensitive dependence on initial conditions, similarity to random behavior and continuous broad-band power spectrum. In such systems, any uncertainty (no matter how small) will produce rapidly escalating and compounding errors in the prediction of the system's future behavior. The highly unpredictable and random-look nature of chaotic signals are attractive features for encryption systems.

According to [6], the main features of a nonlinear dynamical system, exhibiting deterministic chaos for given values of the parameters, are the following:

- Sensitive dependence on initial conditions - where small changes in the initial values of variables grows in time, and produce unpredictable difference as we compute further, the orbit or path.
- Irregular motion in phase space - illustrated by very complex, sometimes noise-like patterns of oscillations of the solutions within a bounded, compact sets. The particularity of chaos is that such complex oscillations are fully reproducible for same numeric precision in the initial conditions and parameter values. Such quasi-stochastic behavior can be qualified by the specific character of the associated measures and invariant densities.
- Qualitative change of the character of the solutions-illustrated by a one or more subsequent bifurcations, structural changes of the phase set to which chaotic solutions converge as we evolve the system in time. Such attractors, i.e. chaotic attractors sometimes do not resemble at all the topological structure of other solutions, for example the period orbits. This is result of a global structural change of the phase space. Compact, simply connected sub- sets, along certain parameter ranges undergo series of non-smooth changes in their geometry and topology, mainly due to subsequent stretching and folding cascades. Such attractors are also called strange attractors, due to their specific geometry and self-similar structure on different time-scales.

Encryption Types: Today, one distinguishes between two fundamentally different approaches to cryptography: symmetric and asymmetric (or public key) mechanisms.

The cryptographic mechanisms is that the sender and receiver of a secured communication use the same key for its protection. Thus for each communication link the communicating partners have to share a common secret key. The main drawback of this approach is the number of necessary keys (i.e. one per communication partner) that have to be stored by each participant and the problem to securely agree on or exchange an appropriate key to set up a secure communication link. Modern symmetric mechanisms are typically based on rather simple operations, like bit shifts and XORs and are therefore quite efficient. The most common application of this class of cryptographic mechanisms is the symmetric encryption.

In symmetric encryption a plaintext message is encrypted by the sender using a secret key. The resulting cipher text can than be decrypted by the receiver with the same (or a closely related) key while non-legitimate persons can not reconstruct the original message from the cipher text without this key. Symmetric encryption hence provides confidentiality of the transmitted information and one of the most popular symmetric cryptographic system is AES (Advanced Encryption Standard).

On the other hand, wit the rapidly increasing number of participants and new applications beyond closed user groups, there are drawbacks of symmetric cryptography concerning key handling and key storage became even more critical. Fortunately, these problems were overcome with the invention of public key cryptography in the 1970s. While in the symmetric setting, the same (secret) key is used for encryption and decryption (as well as MAC generation and verification), these functions are separated from each other in the public key scenario. Here, the secret key is split into a private and a public key part. The private key has still to be kept secret, whereas the public key is published in an appropriate register. The receiver's public key is used for encryption and verification of digital signatures while the private key is used for decryption and signature generation.

AES Advanced Encryption Standard: The National Institute of Standards and Technology (NIST) established a new standard for symmetric encryption procedures: AES, Advanced Encryption Standard. The main objective is the establishment of a Federal Information Processing Standard (FIPS). AES specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext.

The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. For more information please visit [7] http://csrc.nist.gov/publications/PubsFIPS.html

Ordinary Differential Equation (ODE): An ordinary differential equation (ODE) is a mathematical equation containing a function of one independent variable and its derivatives.

Additionally, the reader who wants to have a deeper understanding of the research, we recommend to familiarize with the following topics: current status of the encryption market, chaotic systems synchronization, cyber attacks evolution, computer architecture trends like secure hardware computer architecture and quantum computing, PGP, operating systems compatibility, networking communications and cloud services.

2.4 Research Summary

The overall intent of this research was to build a method and apparatus (hardware, interface & software) to allow *secure communication* between two parties. The firmware provides software-programmable hybrid-encrypting/decrypting hardware realizing chaotic dynamics for the transmitting/receiving devices, as well as a methodology and software to create a secure communication protocol with such devices. The information will be encrypted in the sender device using an ∞-key (infinite key), using information generated from a chaotic oscillator (see Fig. 1). The *cypher* can be transmitted via secure or insecure channels. At the receiver end, similar firmware will use its locally-generated ∞-key to decrypt the received cypher, recovering the original message.

There are many encryption/decryption algorithms and applications available in the market today [7, 8, 9, 10, 11]. Most of them rely on digital schemes to encrypt and decrypt a message. Given enough resources (time, computational power, etc.), these methods can be circumvented and the apparent security compromised. With the advent of quantum computing, the common consensus is that the time needed to crack current encryption would be reduced drastically.

There have been some attempts to create analog encryption/decryption using chaotic oscillators [12], but their reliability is poor for commercial use and once the hardware is breached, the encryption/decryption method is exposed and compromised. In chaotic communication systems, the message's masking is performed at the physical layer by embedding the signal within a chaotic carrier in the emitter. The recovery of the message is based on the synchronization phenomenon by which a receiver, quite similar to the transmitter, is able to reproduce the chaotic part of the transmitted signal.

Our proposed method, uses a combination of mixed-signal hardware and software that is capable of realizing digitally-tunable hybrid chaotic oscillators [13, 12]. This includes the class of chaotic oscillator, hybrid integration reset's thresholds and so on. At the same time, adjustable methods of how to use the chaotic oscillator information to encrypt and decrypt both analog and digital information are developed. This will make the secure information invulnerable to digital information compromises or hardware breach.

In this research, we have developed successfully a software-only prototype of the technology that could be used (as-is) as a potential product to be beta-tested within UT. In addition to the prototype, the long term vision of this project includes hardware components inside a secure hardware-software architecture. For instance, the architecture could be included inside communications devices currently available in the market (e.g., iPhone) in order to offer a new option for data protection. The roadmap also includes the following features:

- An "infinite key" generation algorithm.
- Software-reconfigurable hardware.
- On-demand selection of Chaotic oscillators and mixed-signal processes.
- Cloud-enabled service.

Since our solution ultimately will be implemented in hardware, it will execute faster than current techniques of similar complexity. The several layers of complexity offered by this invention may thwart an attack or convert it into an ineffective endeavor. This research has delivered a crypto-tool that uses keys generated by an integrated piece of software that simulates chaotic oscillators with mixed-signal integrators. Figure 2 shows the correlation between the different concepts involved in this project.

2.5 Accomplished Goals

The project was focused on the demonstration of the components of the technology:

1. Define and create a new chaos-based encryption method using a combination of mixed-signal integrators, chaotic oscillators and standard encryption algorithms.
2. Demonstrate that the technology works with a software prototype and could be available in the market with the hardware component through a multi-year grant (to be proposed as a next phase to another agency or commercial enterprise).

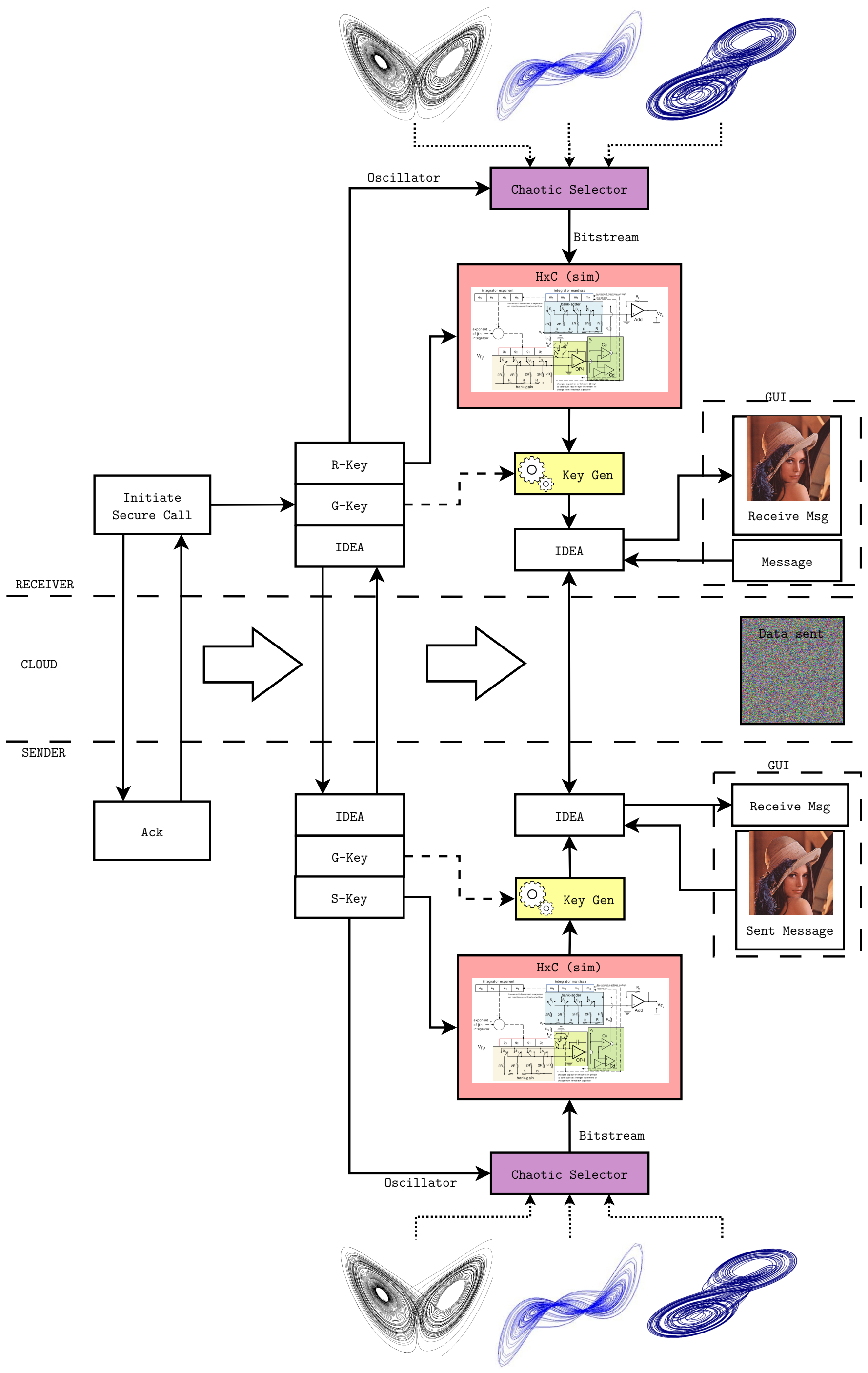


Figure 1: Diagram of CHECK Chaotic Hybrid Encryption Communication Kit. The connection originator places a secure transmision or, while in a transmission, switches to secure mode. The responder establishes a secure protocol to exchange sender and receiver keys that set up the hybrid encryption algorithm. Each user's device spawns the chaotic oscillator simulation in the HxC simulator. During the communication, the sender message is encoded using a sector of the ∞-key and sent over the communication channel. The receiver decodes the encrypted message using the ∞-key stream reproduced locally by its chaotic decryptor.

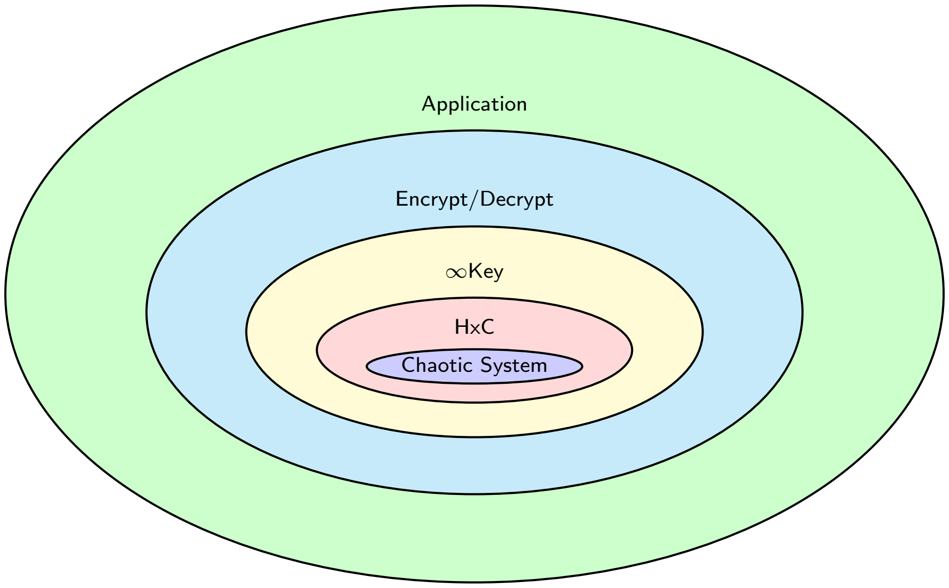


Figure 2: Overview of Software components. Each layer provides additional levels of security. Even if the hardware and/or the encryption algorithm are known, there are still infinite possibilities on the other layers.

2.6 Research questions

The research question that the proposed project was trying to tackle were:

1. Is it possible to create a new encryption method through a modification in the key generation mechanism that generates a chaotic infinite encryption key and that is *unique* to the device?
2. Using our patent-pending technology, is the combination of software-programmable chaotic oscillators and customizable mixed-signal integrators a stable mechanism to generate infinite keys?
3. How secure is the proposed architecture and protocol?

2.7 Objectives

The specific objectives of the proposed project were:

1. Design and develop an infinite-key generation tool using our pending patent: Hybrid eXtreme Encryption Apparatus and Method (HxE)[1].
2. Design and develop a chaos-based crypto-tool that allows to demonstrate `1:1` encrypted communication between two devices in the same network.

[1]Patent application pending.

2.8 Benefactors

Among the possible beneficiaries of the overall project (beyond the scope of this proposal) are the governments of the World, private citizens and corporations. We foresee communication carriers and device manufacturers including this technology in their devices or relaying hardware. Some examples of future use for this research:

- Exchange of confidential documents within a governmental office.
- Data transmission of intellectual property between two researchers at any campus.
- Private consultation between clients and advisor.
- Private phone calls through public networks.

3 Proposed Method

Since the last two decades, we have been living an era where the information is the most important asset to protect for corporations and persons. Moreover, cyber attacks are no longer an exclusive problem of the military or enterprise. As a matter of fact, those types of attacks have been present in the news more frequently than expected. Today, the world is more exposed and we need more and new data protection solutions to mitigate the risk and try to protect private information against attacks. As we mentioned before and with those ideas in mind we have been working in an new encryption method that can be part of a integral solution to accomplish that goal. We have designed the solution around the four principles of the Cryptography:

- *Confidentiality:* A message can not be accessed by no-authorized user.
- *Authentication:* The origin of a message should be available to the receiver.
- *Integrity:* If the message has changed during the transmission, the receiver has to know that.
- *Liability:* A sender should not be able to deny having sent a message.

Taking in consideration the importance of the cryptography in the cyber-security market, the NERDLab (Neuro-Engineering Research & Development Laboratory) has assembled a multidisciplinary team of engineers to apply their knowledge in this particular issue. As a matter of fact, all the engineers who worked in the project have a broad knowledge and international experience in advanced technologies, software architecture and development of enterprise applications. Additionally, we have filled a patent in order to protect the Intellectual Property of our invention. The following paragraphs will show the development phases of the project and their milestones.

3.1 Cryptography Review & Hybrid Encryption

We have conducted an extensive review about the security concepts around modern cryptography. Even before the project started, the whole team had reviewed and tested several concepts around some of the cyber-security challenges in Today's market. It is important to mention, the idea about the integration of the hybrid-processor and the chaotic oscillator to generate infinite keys was conceived before this research. In fact, after we saw the potential of this idea we filed an IP disclosure with UT's Office of Technology Commercialization (OTC) on how the combination of the concepts behind the proposed research could be an innovative method of encryption. This funded project allowed us to dedicate our efforts to validate the ideas by building a software version of the hybrid processor (US Patent 7,555,507), simulating the analog part in software where the errors and almost imperceptible. Indeed, this piece of the architecture is a fundamental concept used during our research, because the hybrid processor will resolve de differential equations for each chaotic system.

The design of the hybrid processor merges analog and digital circuitry to overcome computational difficulties associated with either computer platform. The machine consists of a multitude of interconnected, interacting, mixed-signal integrator cells, (schematically shown in Fig. 3) augmented with a mixed-signal nonlinear approximation function. All integrations and calculations are time continuous and over the set of "real numbers", using a mixed-signal floating-point representation consisting of exponent, mantissa, and *analog bit*$^{\mathrm{TM}}$. The hybrid system is not a sampled-data system, but a true mixed-signal processor. Performance, speed, accuracy, and size will substantially outperform modern supercomputers at a minuscule fraction of the cost, size and power consumption. In addition, the technology permits signal processing without analog-to-digital converters (in the traditional sense). Currently we have a Hardware prototype of the hybrid processor. So, for futures phases of the continuing research we envision a platform that could be embodied as a co-processor, giving a laptop or hand held device supercomputer capabilities.

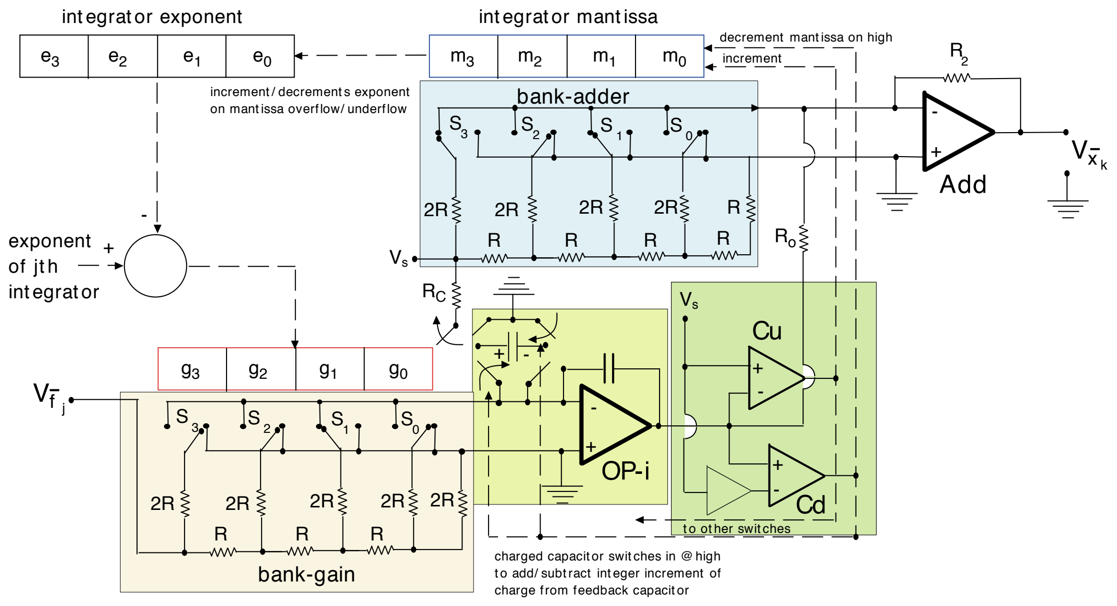

integrator exponent
e3
e2
e1
e0
integrator mantissa
m3
m2
m1
m0
decrement mantissa on high
increment
increment/decrements exponent on mantissa overflow/underflow
bank-adder
R2
Add
Vx̄k
Ro
Vs
RC
exponent of jth integrator
Cu
Cd
g3
g2
g1
g0
Vf̄j
OP-i
2R
R
bank-gain
to other switches
charged capacitor switches in @ high to add/subtract integer increment of charge from feedback capacitor

Figure 3: Hybrid Integrator Cell.

3.2 Architecture and tools

During the second phase of the project, we selected the languages and the software architecture. We settled with two languages for the construction of the prototype: *C++* and *Python*. Python is an excellent language for prototyping, while the code of the core libraries was implemented in C++ looking for a better performance in the most critical part of the prototype. Another advantage of coding the core in C++ is that the code is easy to integrate with different environments using a wrapper. In this case we use *Boost-Python* to build the wrappers. The wrappers allows to call classes implemented in C++ from classes implemented in Python. In fact, all the code associated with the GUI was implemented in Python. Another important design factor to consider was the use of MVC (Model-View-Controller) as a software architecture to implement the application. MVC is a very popular approach, although at this phase MVC is not absolutely necessary, but further down will be extremely helpful in the next development phases. In summary after the evaluation and discard process of different options we have selected the following languages and tools to implement the software:

- C/C++
- Boost
- Boost-Python
- Open SSL
- Python.
- TKinter (Python Library for GUI)
- Sockets TCP/IP with IPV4
- Binary Data Transmission

3.3 Software development

On the third phase, we started with a software simulation of the hybrid processor. We developed a library in C++ called *HxCOdeInt* that simulates the expected behavior of the hardware version of the hybrid processor.

`HxCOdeInt` is a library for solving initial value problems (IVP) of ordinary differential equations (ODE) using the hybrid integrator (HxC). Numerical approximations for the solution $x(t)$ are calculated using a fourth-order*Runge-Kutta* method, available in *Boost*.

Additionally, our library provides a set of pre-configured system of equations that can be used by the solver. The available system of equations are: Chua's oscillator, Duffing's oscillator, Lorenz's oscillator, Vanderpol's oscillator, and basic harmonic (RC) oscillator. The library also provides a set of control methods to define the integration parameters: thresholds, initial conditions, and length of simulation.

The library also includes a class called `HxCOdeIntTest`. The class is responsible for the testing process of the HxCOdeInt library. In fact, with its methods we can perform different types of testings: unit-testing and full-system integration test. The full-system integration test generates plots and data (text files) to keep track of the performance of the library.

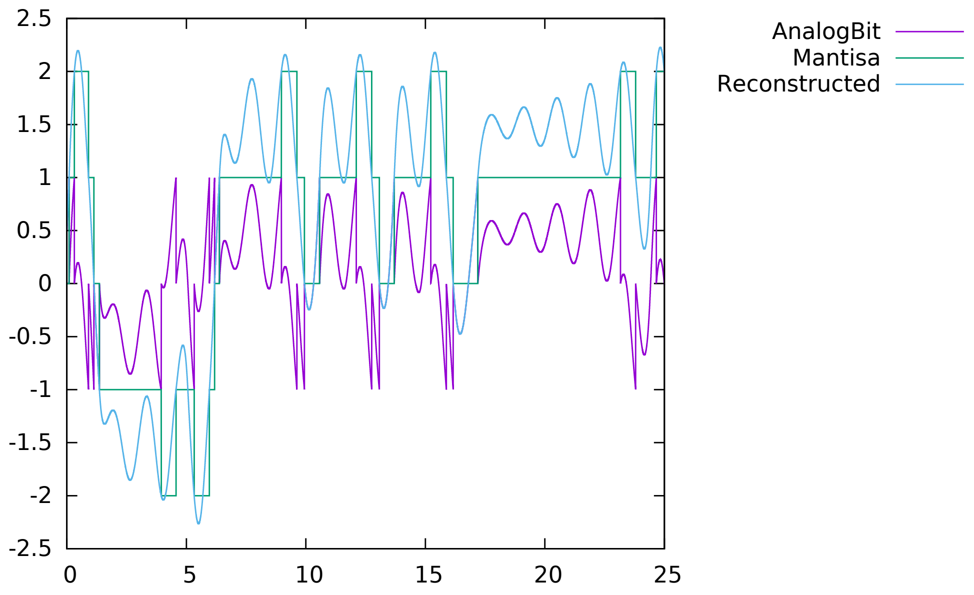


Figure 4: Hybrid Processor simulation 1.

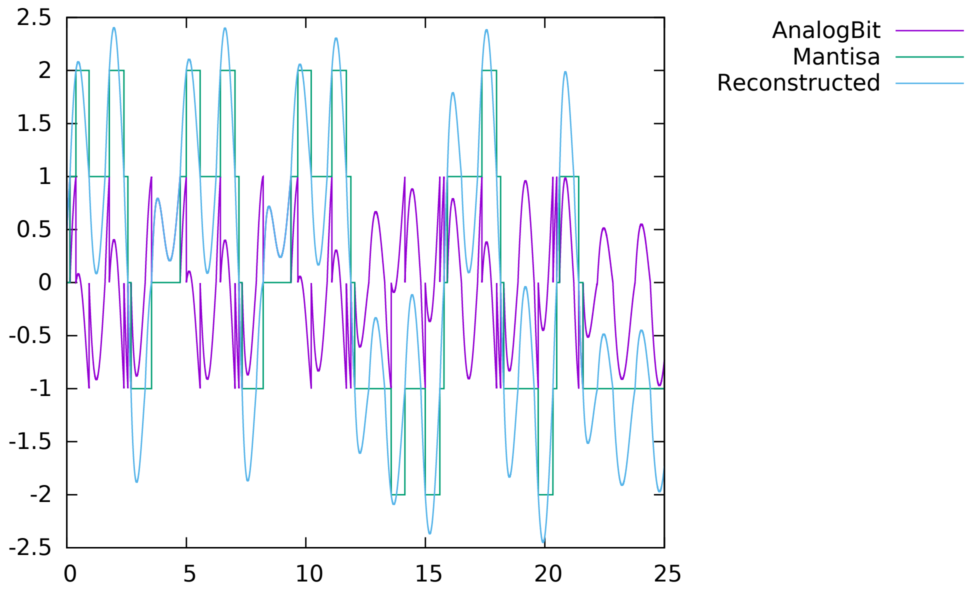


Figure 5: Hybrid Processor simulation 2.

Figures 4-5 provide images are plots generated by the HxCOdeInt Library. Figure 6 shows plots generated by the HxCOdeInt after the resolution of a chaotic system. Figures 7-8 are results of the chaotic systems generated by our simulator.

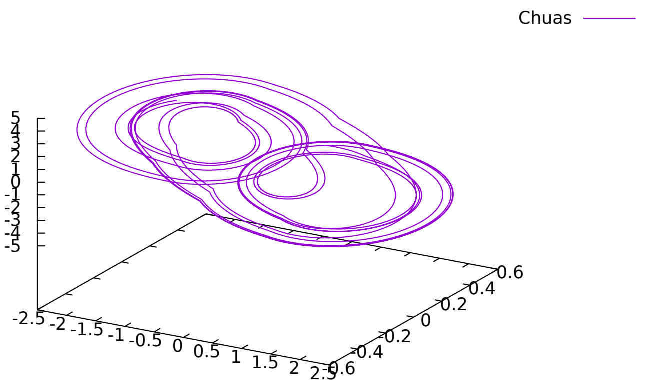


Figure 6: Hybrid Processor Chuas Oscillator.

The next step during this phase was the development of two additional libraries: `HxCKeyGenerator` Library and `HxCEncryptor` Library.

HxCKeyGenerator: Symmetric encryption requires a key that is the same for the encrypting and for the decrypting party. This library is responsible to generate the encryption/decryption keys continuously. To achieve this goal, the class HxCOdeInt uses one (1) chaotic system. The solution of the ODE is processed to generate a bit stream (binary data) and divided in specific length keys.

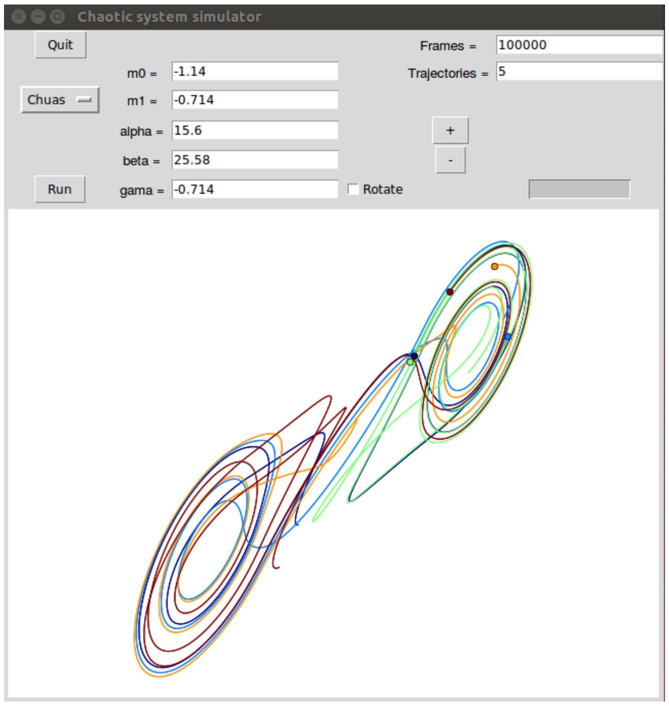
HxCKeyGeneratorTest: This class is responsible for testing the HxCKeyGenerator library. It performs tests to different methods of the class (unit testing) and also performs an full-system key-generation test. The full-test generates several keys from the ODE solution of the chaotic system. The keys generated are saved in text files to keep track of the performance of the library.

HxCEncryptor: This library provides a chaotic symmetric algorithm for encryption and decryption. As a matter of fact, encryption and decryption are often very similar operations for symmetric algorithms. That is reflected by not having to choose different classes for either operation and both can be executed using the same class HxCEncryptor.

Once the instance of the HxCEncryptor is obtained, the system has to configure: the hybrid integrator parameters, the chaotic system to be used and the key generator length. Moreover, the configuration process must be the first call after creating the class instance.

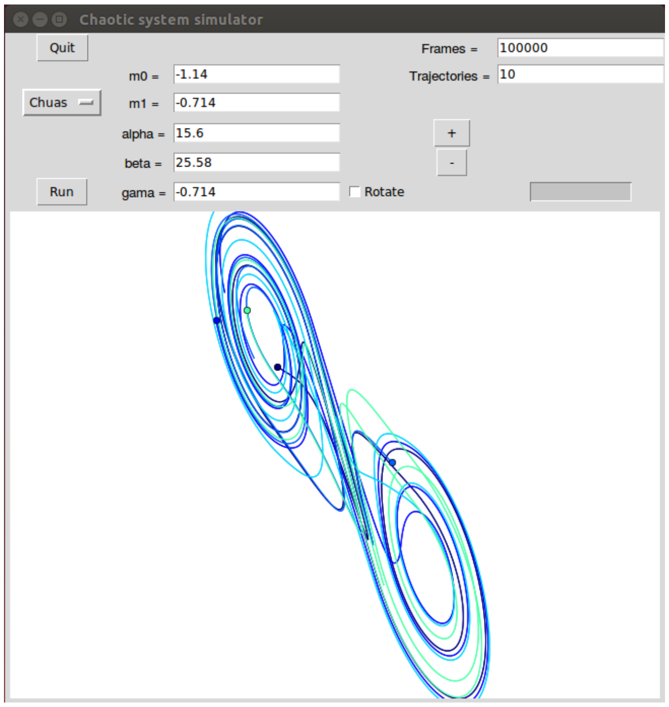
The Encryption process consists of the following three stages:

1. Setting up a context: configure hybrid integrator parameters, select chaotic system to use and setting key length.
2. Initializing the encryption operation: run ODE solver and convert the solution to binary data, this process generates the new encryption key.



Frames = 100000
Trajectories = 5
m0 = -1.14
m1 = -0.714
alpha = 15.6
beta = 25.58
gama = -0.714
Rotate

Figure 7: Chuas Chaotic Oscillator Simulation.



Frames = 100000
Trajectories = 10
m0 = -1.14
m1 = -0.714
alpha = 15.6
beta = 25.58
gama = -0.714
Rotate

Figure 8: Chuas Chaotic Oscillator Simulator.

3. Finalizing the encryption operation: to complete the encryption we provide the plaintext bytes to be encrypted. The encryption is made using AES 256 with the recently created encryption key.

Then, we need to define the *Decrypt* operation. The decryption operation, is similar to the encryption and consists of the following stages:

1. Setting up a context: configure the hybrid integrator parameters, select the appropriate chaotic system to be used and setting key length.
2. Initializing the decryption operation: run the ODE solver and convert the solution to binary data, providing a *cypher*'s text bytes to be decrypted.
3. Finalizing the encryption operation: the decryption operation receives the cipher text as input and returns the recovered *plaintext*. In the current implementation, the decryption is made using AES 256 using the recently created encryption key.

In order to obtain a accurate result after running every encrypt - decrypt iteration, both chaotic systems (encryptor and decryptor) must be *synchronized*. Furthermore, the system execute an auto reconfig after a given number of iterations.

HxCEncryptionTest: This class performs a full-system test of the system executing an encryption and decryption cycle. During the test, two instances of the `HxCIntegrator` are created, one configured as encryptor and the other one configured as decryptor. The test consist in the following steps:

First, the systems are configured to solve the same ODE system with the same parameters (thresholds, IC, etc.). Second, the output of the ODE solution is then used internally to generate the encryption and decryption keys. Third, a text message is encrypted and decrypted. Lastly, The test compare the generated message from the decryption with the original message.

We have considered two types of encryption: symmetric and asymmetric also know as public key. However, we chose AES as a symmetric encryption type in order to do the testing of our tool.

Graphical User Interface: We have developed a graphical user interface (GUI) and a set of wrappers using boost-python capabilities to allow calls to the previous libraries from the GUI. The idea of the wrapper was to maintain the core libraries available for future integration with other languages or user interfaces like mobile. Another important advantage from this software architecture is that the design is already considering that some libraries could be replaced by a Hardware component. In future versions, the hardware component will be the Hybrid Processor, which is currently a prototype as well.

The GUI was developed using TKinter which is Python's de-facto standard GUI package. For future versions of the tool we will consider other Libraries with better graphics quality, widgets and native integration to different operating systems. The same assumption applies to the mobile version. In fact, for the mobile version we will use another wrapper technology to execute calls to the encryption library from an app.

The following figures are screenshots of the GUI:

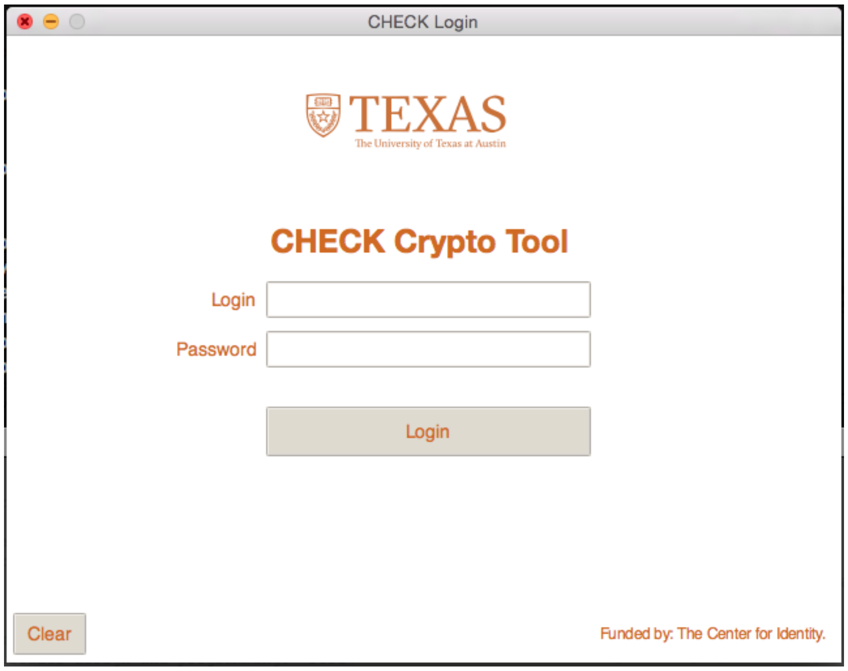
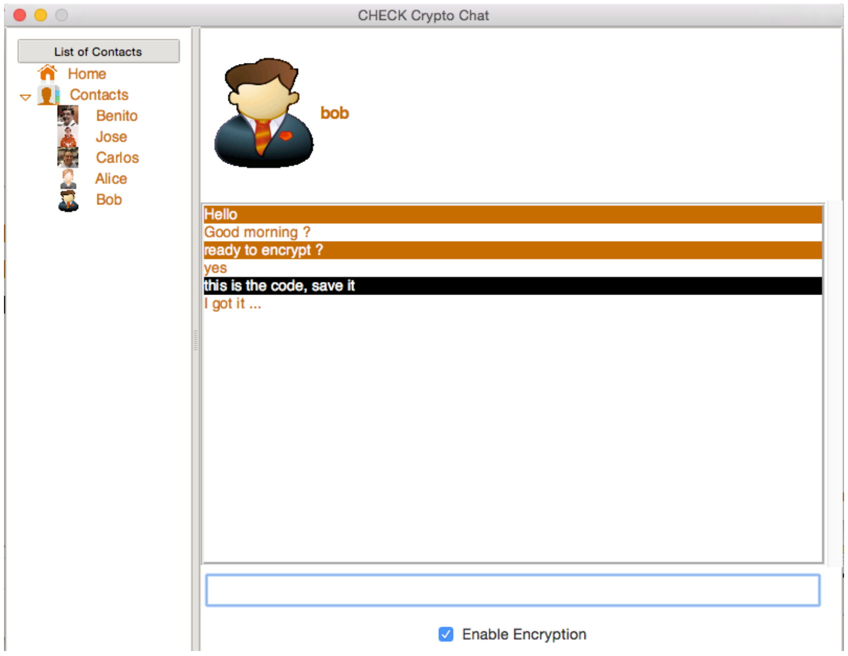

CHECK Login
TEXAS
The University of Texas at Austin
CHECK Crypto Tool
Login
Password
Login
Clear
Funded by: The Center for Identity.

Figure 9: CryptoChat login window.



Hello
Good morning ?
ready to encrypt ?
yes
this is the code, save it
I got it ...

Figure 10: CryptoChat main window.

3.4 Continuous Testing

Our methodology executes continuous testing and quality assurance of the crypto-tool. As we mention before, for each milestone of the project we use a *Test Class* to validate that the results were correct. Additionally, we have prepared two proprietary scripts during the testing process: one for key generation and another for encrypt/decrypt testing. The intention of the first script was validate if the key generation library was working correctly before to continue with the next phase of the project.

As a matter of fact, this part of the testing was very important because we introduce in this research the feature of *infinite key* in our tool. So, we must be sure that the key generation mechanism was accurate and reliable before to use our keys with AES.

Once we finished the key generation test, we prepared an script to test consecutive calls of the encryption/decryption process before to integrate that Library with the GUI. In this step, we have tested the script with 100 calls and then with 1000 consecutive calls. In both cases with got zero errors.

3.5 Team members

This proposal involves 3 researchers with expertise in different disciplines.

Benito R. Fernández, Ph.D., M.S. Mech. Eng, B.S. Chem. Eng. & Matls. Engineering. Expert in computer science and mechatronics, will be a part-time advisor and the principal investigator. During the whole duration of the project will be assisting in the design and development of the hardware-software architecture.

José Rafael Capríles, Ph.D. student, M.S. Automated Control & Robotics and B.S. Electronic Engineering. His expertise includes electronics circuits design and programming. Jose's Ph.D. research is in the hybrid integrator development.

Carlos Augusto García, B.S. Computer Science Engineering. Advanced expertise in Storage Technologies, Unix Base Operating Systems and Networking solutions. Carlos has more than 15 years of experience working with complex technologies and IT solutions for large accounts in industry.

4 Summary of results

During this phase of the project, our team have accomplished the following goals:

- HxC simulator software version.
- Chaotic oscillator software simulator.
- Key Generator tool.
- Encrypt / Decrypt Library.
- Testing scripts for Encrypt/Decrypt tools.
- Crypto Chat graphical user interface.

5 Future steps

We encourage the Center for Identity to consider assigning additional funds in order to continue with this project. The suggested steps for further research should be:

- Write journal and conference papers after the patent is filed and the OTC (Office of Technology Commercialization) clear us for publication. We also plan to submit at least one proposal to a government agency (possibly DARPA, ONR, AFOSR, or NSF) and/or secure funding from industry. An important application that we foresee is to use it in Cyber-Physical Systems (CPS).
- Secure funding for continue with our research.
- Intensive testing trying to break our method.
- Cyber attacks simulation for testing purposes.
- Design, develop, and build the Hardware version of the Integrator. This include manufacture a board version of the HxCOdeInt using discrete components. However the long term vision of this phase is to build an IC.
- New version of the method (hardware,software and Interface), using the hardware version of the HxCIntegrator .
- New software versions will include: user access control, cloud services, networking services and a new interface.
- Mobile/web version of the Tool.

References

[1] S. Kandola, “A survey of cryptographic algorithms,” Ph.D. dissertation, St. Lawrence University, 2013.

[2] N. Moshopoulos and E. Chaniotakis, “A survey of cryptography algorithms–trends and products,” *HeroonPolytehneiou*, vol. 9, no. 15773, 2000.

[3] M. Bryant, A. Seth, B. Fernández *et al.*, “Apparatus for solving differential equations,” Jun. 30 2009, uS Patent 7,555,507.

[4] B. Remy, M. D. Bryant, B. R. Fernández, S. Yan *et al.*, “Mixed-signal system for performing taylor series function approximations,” Nov. 18 2008, uS Patent 7,454,450.

[5] B. R. Fernández and Z. Wu, “Method and apparatus for internally calibrating mixed-signal devices,” Sep. 14 2010, uS Patent 7,796,075.

[6] D. Solev, P. Janjic, and L. Kocarev, “Introduction to chaos,” in *Chaos-Based Cryptography*, ser. Studies in Computational Intelligence, L. Kocarev and S. Lian, Eds. Springer Berlin Heidelberg, 2011, vol. 354, pp. 1–25. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20542-2_1

[7] F. I. P. S. P. 197. (2001, November) Advanced encryption standard (aes). [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[8] J. Nakahara, Jr., B. Preneel, J. Vandewalle, and J. V, “A note on weak keys of pes, idea and some extended variants,” 2002.

[9] S. Garfinkel, *PGP: Pretty Good Privacy.* O’Reilly Media, 1994.

[10] P. Zimmermann and A. Johnston, *ZRTP: Media Path Key Agreement for Unicast Secure RTP*, zfone project ed., A. J. Callas, Ed. Apple, Inc., April 2011.

[11] J. Daemen and V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard.* Springer, 2002.

[12] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications.* Berlin: Springer, 2011.

[13] M. D. Bryant, S. Yan, R. Tsang, B. Fernández, and K. K. Kumar, “A mixed signal (analog-digital) integrator design,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 7, pp. 1409–1417, 2012.

The University of Texas at Austin
Center for Identity

For more information on Center for Identity research, resources and information, visit **identity.utexas.edu.**