DESIGN AND IMPLEMENTATION OF CRC BASED CRYPTOGRAPHY USING FPGA FOR SECURED DATA COMMUNICATION

Chethana R, Rakshitha L, Sumalatha R, Vaishnavi R Vedhavyas

Electronics and Communication Engineering, Dr Archana HR Assistant Professor BMS College of Engineering, BMS College of engineering, Basavanagudi, Bangalore, India.

Chethana.ec20@bmsce.ac.in, rakshitha.ec20@bmsce.ac.in, sumalatha.ec20@bmsce.ac.in, Vaishnavi.ec20@bmsce.ac.in

Abstract. *For data transmission accuracy, combine CRC error detection with AES encryption. In this method, cypher text is produced by first encrypting the data with AES. The cypher text is next subjected to a CRC checksum calculation, yielding a tiny value that is attached to the end of the cypher text. The message is then transferred or stored in its entirety, including the checksum. The recipient of the message must first decrypt it using the same AES key that was used to create the message's original plain text. Then they compare the decrypted message's CRC checksum to the checksum that was originally transmitted with the message. The recipient can be sure that the message was sent correctly if the two checksums match. If the checksums do not match, the receiver knows that an error has occurred during transmission or storage of the message. The receiver is aware that an error occurred during message transmission or storage if the checksums do not match. In the AES Design unit, CRC is utilised in place of standard Key Generation to create keys. TRNG (True Random Number Generators) replacement has improved performance in terms of area and delay. Data encryption is now substantially more secure thanks to the enhanced encryption standard algorithm. To create a random integer or key for use in AES encryption or decryption, CRC is used in AES. dependable. error-free. This Design is implemented using Verilog HDL and simulated by Xilinx Vivado and synthesized by Xilinx tool. CRC can be used in cryptography to ensure that encrypted data is transmitted without corruption and that the resulting decrypted data is accurate and reliable.*

1. *Introduction*

In digital networks and storage devices, a cyclic redundancy check (CRC) error-detecting algorithm is widely employed to spot unintended changes to digital data. In these systems, the remaining polynomial division of each block of data that enters determines a brief tick value. When the data is retrieved, the computation is run so that, in the event that the tick values do not match, data corruption can be prevented. Errors can be fixed with CRCs.

Because the algorithm relies on cyclic codes and the check (data verification) value is redundant (increasing the message without adding information), CRCs are named for this property. CRCs are popular because they are easy to implement in binary hardware, uncomplicated to assess analytically, and particularly effective at identifying frequent errors caused by noise in transmission channels. Because it has a specified length, the function that generates the check value is occasionally employed as a hash function.

Cryptography is a method for securing data and communications through the use of codes, ensuring that only the intended audience can decipher and process them. preventing unauthorised individuals from accessing information. The prefix "crypt" (which means "hidden") and the suffix "s" (which means "writing") are attached to the terms "crypt" and "graphs," respectively. The techniques employed in cryptography to secure data are drawn from mathematical concepts and a collection of rule

-based calculations known as algorithms to modify communications in a way that makes them challenging to decode. These algorithms are used for a variety of tasks, including the development of cryptographic keys, the formation of digital signatures, the protection of data privacy, online browsing, and the security of private transactions

like debit and credit card payments.

Cryptography techniques: In the age of computers, cryptography is commonly associated with the conversion of plain text into cypher text, which is text that can only be deciphered by the intended recipient. Encryption is the term for this procedure. The process of transforming encrypted text into plain text is called decryption.

A block cypher that uses an encryption key and numerous rounds of encryption is the AES encryption algorithm. A block cypher, a type of encryption method, only encrypts one block of data at a time. The block size in standard AES encryption is 128 bits, or 16 bytes. AES is not a standalone computer software or piece of source code. It is a justification in mathematics for how to conceal data. Several people have incorporated AES into their source code.

During the encryption procedure, AES encryption uses a single key. The key's size might vary between 128 bits (16 bytes) and 256 bits (32 bytes). The term "128-bit encryption" refers to the use of an encryption key of that many bits. With AES, the encryption and decryption processes use the same key. This is a symmetric encryption algorithm. The term "asymmetric encryption algorithms" refers to those that operate using two different keys, a public key and a private key. The encryption key is a binary string of information utilised during the encryption process. Because the same encryption key is used to encrypt and decode data, it is essential to employ encryption keys that are challenging to decipher. Some keys are generated by software used for this specific activity. An alternate strategy is to create a key out of a passphrase. In effective encryption systems, a pass phrase is never used as the only encryption key.

This approach is helpful because it adds an additional layer of error detection to ensure the integrity of the sent data in circumstances where data transmission or storage is unreliable, such as across noisy channels or in wireless communications. It's important to keep in mind that CRC provides no additional security beyond what AES already provides. In this case, AES serves as the primary security mechanism, and CRC serves as an additional layer of error detection.

2. *Proposed work*

Segmented CRC:

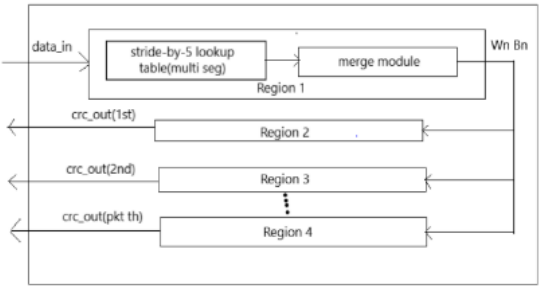


Fig 4.1.1. Segmented CRC

Short or out-of-alignment frames have a lower throughput due to the non-segmented system architecture's inability to execute several frames in a single word (clock). This is referred to as the bus efficiency issue. To address the issue, a segmented system architecture is suggested. The block is another name for the segment, and the bus format is the same as that. For instance, a 4096-bit bus can divide into eight zones since it can execute eight full frames at once. The only factor affecting the number of regions is bus width. A region can be divided into eight segments (or "blocks") if a 64-bit segment width is used, while other segment widths are possible. The picture above depicts the suggested segmented system architecture. The proposed segmented system architecture features several duplicates of Regions 3 and 4 and a little more complex Region 1 and Region 2 than the proposed non-segmented system architecture.

Non Segmented CRC:

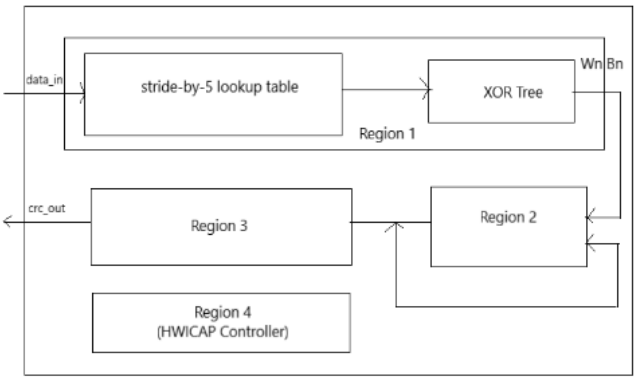
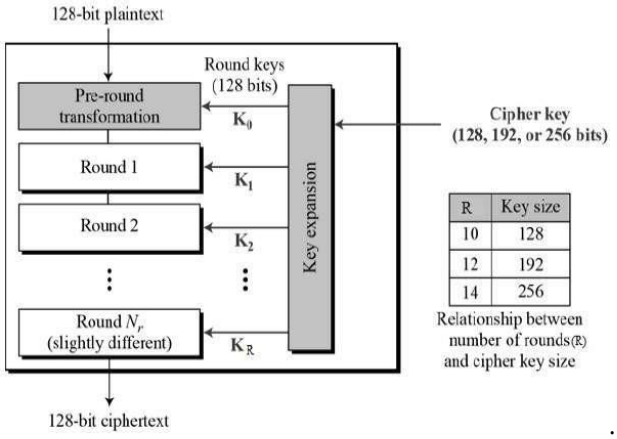


Fig.4.1.2. Non-Segmented CRC

The suggested non-segmented system architecture is shown in Fig. A non-segmented system architecture should only have one frame per word while a segmented system design can handle several frames at once. WlnBn is calculated in accordance with Regions 1 and 2. Region 1 consumes the majority of LUTs, and the amount consumed is inversely associated with Wln size. The stride-by-5 technique is recommended to reduce the LUT usage in Region 1. Region 2 is built utilising a xor tree rather than a one-stage XOR algorithm to achieve higher performance. Region 3 completes the calculations in (1). Tn employs few LUTs because of its small size. The padding zeros problem is resolved in Region 4, and the pipelining go back strategy is recommended as a fix. Resources used by this algorithm are O (log2 n). The LUTs' content can be dynamically altered using a HWICAP controller called Region 5.

4.1.3. AES Algorithm:

The schematic of AES structure is given in the following illustration



R	Key size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size

4.1.3. AES Algorithm

The AES encryption and decryption process is shown in the above graphic. The algorithm starts with the Add round crucial stage. The remaining rounds consist of nine rounds with four stages each and a final round with three stages. This is true for both encryption and decryption, with the exception that the decryption algorithm step for each round is the opposite of the stage it corresponds to for encryption. The four stages are as follows:

1. *Swap out bytes*
2. *Row-switching*
3. *Columns that mix*
4. *Include Round Key*

The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

1. *Inverse Shift rows*
2. *Inverse Substitute bytes*
3. *Inverse Add Round Key*
4. *Inverse Mix Columns*

AES is a symmetric block cypher. This shows that the same key is used for both encryption and decryption. But there are several ways in which AES and DES are very different from one another. The Rijindael algorithm supports a number of block and key sizes in addition to the 64 and 56 bits of DES. The block and key can really be chosen from a range of 128, 160, 192, 224, or 256 bits without having to match. However, in accordance with the AES standard, the algorithm can only accept keys with a block size of 128 bits and a choice of three distinct key lengths: 128, 192, or 256 bits. Depending on whatever version is being used, the name of the standard is modified to AES-128, AES-192, or AES-256. In addition to these features, AES differs from DES in that it is not a feistel structure. Keep in mind that a feistel structure first modifies one half of the data block before switching the other half. Permutations and replacements are utilised in this illustration to process the entire data block concurrently during each cycle.

The Pipelined algorithm is a symmetric iteration block cypher. The block and key can each be 128, 192, or 256 bits long. The NIST mandated that the AES develop a symmetric block cypher with a block size of 128 bits. This restriction prevents variations of Pipelined that work on bigger blocks from being included in the official standard. A variable number of rounds or iterations is also offered by Pipelined: 10, 12, and 14 for keys with lengths of 128, 192, and 256, respectively. The transformations in Pipelined see the data block as a rectangular array with four columns and four-byte vectors. Another fundamental idea is a rectangular array of 4-byte vectors; the number of columns depends on the key length.

Pipelined decryption entails doing the opposite modifications from those employed in encryption in the reverse order. The rounds' inverses are added after the initial data/key addition, which is its own inverse, and so on until the final round's inverse triggers the start of the decryption process. AES only ever employs bytes for computation, never bits. AES treats a plaintext block's 128 bits as 16 bytes as a result. These 16 bytes are arranged into four

columns and four rows for matrix processing. Unlike DES, the number of rounds in AES varies and is determined on the key size. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Each of these rounds uses a unique 128-bit round key that is obtained from the first AES key. Numerous aspects of the AES are impacted by the key's length. For example, there are 10 rounds for keys with 128 bits, 12 rounds for keys with 192 bits, and 14 rounds for keys with 256 bits. The most common key size that will probably be used is the 128 bit key. Consequently, this AES algorithm explanation describes this particular implementation.

The following features of Rijindael were intended to have them:

- *Resistance to all known attacks;*
- *Speed and code compactness on a variety of platforms.*
- *Simple Designs*

4.1.4. Multiplier:

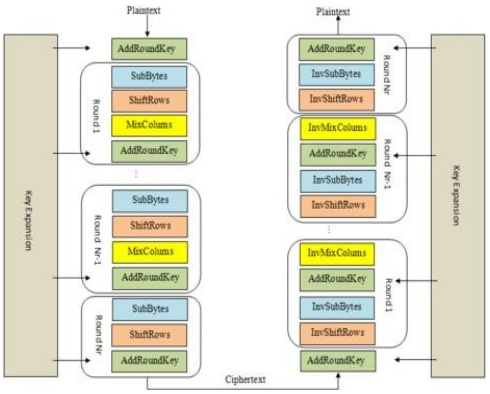
Just a few examples of sensitive and sophisticated infrastructures that are secured by cryptographic systems include secure healthcare, the smart grid, fabric, and homes. Cryptography and the sciences of cryptology and cryptanalysis are closely related. The methods employed in cryptography include employing microdots, combining words and images, and other means of disguising information while it is being stored or transported. The act of converting plaintext (common text, also known as clear text) into cypher text and then back into plaintext (decryption) is how cryptography is most widely understood in today's computer-centered culture. Cryptographers are those who carry out this type of work. However, using cryptographic designs does not make you immune to problems with these infrastructures. Due to shortcomings in VLSI systems, smart usage models might not function well. There has been a great deal of study done on the discovery of these weaknesses in elliptic curve encryption and the Advanced Encryption Standard. The aforementioned fragile cryptographic structures are designed for dependability and fault immunity to ensure their reliability even in the presence of errors.

4.1.4.1. Substitution Box (S- Box):

The Sub Bytes operation substitutes a nonlinear byte. The substitution box, also referred to as the S- box, decides which input state bytes should be substituted for one another. In the finite field GF (28), the S-box is calculated using a bitwise affine translation and a multiplicative inverse.

3. *Proposed system flow.*

The CRC is a common type of error detection code used in digital communication systems to locate errors in data transfer. AES uses the CRC to ensure that the data being transmitted is accurate. In a segmented CRC design, the message is split into fixed-length segments, with a CRC calculated separately for each segment. Error identification and repair are considerably easier because defects can be located and rectified inside each segment separately. In a non-segmented CRC technique, the CRC is computed over the entire message as a single block. Despite being simpler and needing less processing, this technique may be less successful



at spotting faults because errors that occur in one part of the message may influence the entire CRC. Fig 3.1:Proposed system flow.

The method of AES encryption and decryption is shown in the diagram below. An AES cypher text's decryption procedure resembles its encryption procedure in reverse. The four processes are divided into rounds. Add round key

- *Mix columns*
- *Shift rows*
- *Byte substitution*

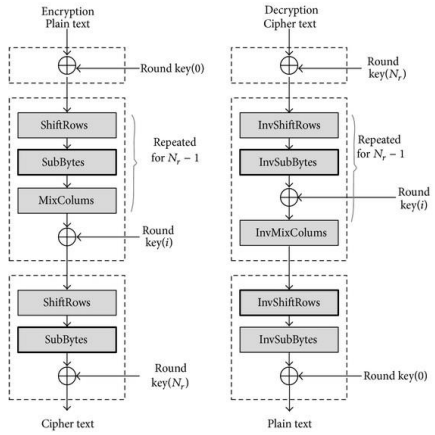


Fig 3.2:Proposed system techniquies

4. Tables.

The table shows the difference between Non Segmented CRC and Segmented CRC. It shows the comparison between the LUT, Slices, gate counts, Overall delay etc.

***Tab.1:** Area and Delay of non-segmented CRC design and segmented CRC design*

	Area			**Delay**		
	LUT	*Slices*	*Gate Counts*	*Overall Delay*	*Gate Delay*	*Path Delay*
Non Segmented CRC Design	*53*	*36*	*8867*	*7.165ns*	*6.364ns*	*0.801ns*
Segmented CRC Design	*66*	*40*	*9001*	*7.165ns*	*6.364ns*	*0.801ns*

4. Conclusion

In order to ensure the integrity of encrypted data during transmission, the CRC algorithm is employed in cryptography. We use CRC error detection in conjunction with AES encryption to ensure the accuracy of sent data. In order to select the best CRC design for TRNG production, we evaluate segmented and non-segmented CRC designs. The overall suggested approach keeps the original algorithm while speeding up and using less space.

Acknowledgements

The writers might express their gratitude in this section of the paper to the projects or individuals who contributed to the findings that were reported in the article. No graphics or logos may be inserted in this area.

Author Contributions

Chethana R and Vaishnavi R Vedhavyas developed the theoretical formalism, performed the analytic calculations and performed the numerical simulations. Both Chethana R, Rakshitha L, Vaishnavi R Vedhavyas, and Sumalatha R authors contributed to the final version of the manuscript. Sumalatha R and Rakshitha L supervised the project.

References

[1]. Raghunath B H;Aravind H. S., "An Efficient FPGA-Based Dynamic Partial Reconfigurable Implementation", IJISAE,2023

[2] Aprilia Putri Dewanty; Bheta Agus Wardijono, "Analysis and Design of CRC-32 IEEE 802.3 Generator for 8 Bit Data Using VHDL", KILAT ,2022

[3] Noor Munir, Majid Khan, Tariq Shah, Ammar S. Alanazi , Iqtadar Hussain, "Cryptanalysis of nonlinear confusion component based encryption algorithm", Science direct, 2021

[4] Huan Liu, Zhiliang Qiu, Weitao Pan, Member, IEEE, Jun Li, Ling Zheng, and Ya Gao, "Low-Cost and Programmable CRC Implementation Based on FPGA", IEEE Transactions On Circuits And Systems, 2021

[5] Favin Fernandes, Gauravi Dungarwal, Aishwariya Gaikwad, Ishan Kareliya, Swati Shilaskar, "VLSI Implementation of Cryptographic Algorithms & Techniques: A Literature Review", IEEE, 2019

[6] Behrouz Zolfaghari, Mehdi Sedighi and Mehran S. Falah, "Designing programmable parallel LFSR using parallel prefix trees", Journal of Engg. Research, 2019

[7] Peter Orosz, Tamas Tothfalusi, Pal Varga, "FPGA-Assisted DPI Systems: 100 Gbit/s and Beyond " , IEEE Communication Surveys and Tutorials, 2019

[8]. R. Roman, P. Najera, and J. Lopez, "Securing the Internet of things," Computer, Sep. 2018.

[9]. Kizzepatta Vigin, Nazarbayev University, Kazakhstan Suhaib A. Fahba, University of Warwick, United Kingdom, FPGA Dynamic and Partial Reconfiguration: A Survey of Architectures, Methods, and Applications, ACM Computing Surveys, Vol. 51,
No. 4, Article 72.,2018

[10] M. Mozaffari-Kermani, A. Reyhani-Masoleh, "Concurrent structure independent fault detection schemes for the Advanced Encryption Standard," IEEE Trans. Comput., May 2017.

[11]S.C. of Low-Cost and Programmable CRC Implementation. Accessed: Apr. 24, 2020. [Online].Available:https://github.com/FPGANetworking/Low-Cost-and-Programmable-CRC

[12] P. Orosz, T. Tóthfalusi, and P. Varga, "FPGA-assisted DPI systems: 100 Gbit/s and beyond," IEEE Commun. Surveys Tuts., vol. 21, no. 2, pp. 2015–2040, 2nd Quart., 2019.

[13] M. Jubin and T. Nayak, "Reconfigurable very high throughput low latency VLSI (FPGA) design architecture of CRC 32," Integration, vol. 56, pp. 1–14, Jan. 2017.

About Authors

***Chethana R** was born in Bangalore. She is pursuing BE from BMS college ofEngineering.*

***Rakshitha L** was born in Bangalore. She is pursuing BE from BMS college ofEngineering.*

***Sumalatha R** was born in Bangalore. She is pursuing BE from BMS college ofEngineering.*

***Vaishnavi R Vedhavyas** was born in Bangalore. She is pursuing BE from BMS college of Engineering.*

Appendix A
AC Driver Parameters

- $P_N = 3kW$,
- $U_{1N} = 220V$,
- $I_{1N} = 6.4A$,
- $n_N = 1420rev{\cdot}min^{-1}$.