Fuzzy Systems and Soft Computing
ISSN : 1819-4362

A STUDY OF PIPELINE TECHNIQUES IN OPTIMIZING AES ALGORITHM

Ms.Shaima Mateen Thange, Assistant Professor, Department of Information Technology,
SIES (Nerul) College of Arts, Science and Commerce (Autonomous)
shaimathange@sies.edu.in

ABSTRACT:
This study investigates how the pipeline technique can be applied to the Advanced Encryption Standard(AES) algorithm to improve encryption efficiency. AES is extensively utilized for secure communication, yet its computational complexity may lead to delays in high-speed applications. The pipeline technique permits multiple stages of encryption to be processed concurrently, lowering latency and enhancing performance. This paper examines both hardware and software implementations of pipelined AES, assesses performance metrics, and discusses the associated challenges. Experimental findings indicate that implementing pipelining significantly accelerates AES processing speed while maintaining security.

Keywords:
AES, Pipeline Technique, Encryption, Hardware Acceleration, Performance Optimization.

INTRODUCTION:

Advanced encryption standard(AES)

The AES encryptional gorithmissymmetric in the group, and there are three different key lengths: 128 bits, 196 bits, and 256 bits, with the packet size being 128 bits. The algorithm is reasonably flexible in its application. The AES algorithmis widely used in software and hardware. Inthethree keylengths, the 128bit keylength is commonlyused. The internalalgorithmperforms a ten-time iterative process when the key length is under. The five sections of the final round are joined by the Sub Bytes, S-box, Shift Rows, Mix Columns, and Add Round Key. AES has five different unitsofmeasurement:bits, bytes, characters, groups, states. AroundofAESiscomposedofbyte replacement (Sub Bytes), line displacement (Shift Rows), mixed column displacement (Mix Columns),keyreplacement(AddRoundKey),andsoon.AESalgorithmdesignshouldmeetthree criteria during all phases of the data packet.

SubBytes: Byte substitution using a predefined S-box.
ShiftRows: Row-wise Permutation of data.
MixColumns: Column-wise mixing of data.
AddRoundKey: XOR operation with a round-specific key.

Mathematically, AES transformations can be expressed as:
SubBytes: $S'(x) = S(x)$, where $S(x)$ is the substitution table.
ShiftRows: $R'(i,j) = R(i,(j+c[i]) \bmod N)$, where $c[i]$ represents the row-dependent shift.
MixColumns: $M'(x) = M \cdot x$, where M is the mixing matrix and x is the column vector.
AddRoundKey: $A'(x) = x \oplus k$, where k is the round key.

LITERATURE REVIEW:

Optimization methods for AES encryption have been the subject of numerous investigations. Pipelining is used in hardware implementations, like FPGA-based AES, to increase encryption speeds. To increase throughput, software-level pipelining in multi-core CPUs has also been studied. This section examines earlier studies on pipelining's function in cryptographic systems and how to improve AES performance.

PIPELINE TECHNIQUEIN AES ALGORITHM:

AES encryption consists of several cycles, including Sub Bytes, Shift Rows, Mix Columns, and Add Round Key transformations. The pipeline technique partitions these operations into separate steps, allowing for parallel execution of different cycles. Each step of the AES algorithm can be mapped to an individual pipeline stage in hardware or executed in parallel threads in a software implementation.

HARDWARE PIPELINE:

In an FPGA or ASIC implementation, each AES transformation (Sub Bytes, Shift Rows, Mix Columns, Add Round Key) is assigned to adedicated pipeline stage. Encryption performance is improved because multiple encryption operations are performed in parallel.
This methods ignificantly reduces processing time, making AES encryption suitable for real-time applications such as VPNs and secure messaging.

SOFTWARE PIPELINING:

Inamulticoreprocessor,theAEStransformationscanbesharedacrossdifferentprocessingcores. Techniques such as loop unrolling and instruction-levelparallelismallow AES to run efficiently without requiring specialized hardware. Cloud computing and distributed encryption services benefit from software pipelining, which processes multiple encryption requests simultaneously.

NETWORK PIPELINING:

SecurecommunicationprotocolslikeTLSandSSLuseAESpipeliningtoencryptmultiplepackets at once.
Reduces network latency and improves security performance in real-time applications such as online banking and video conferencing.

METHODOLOGY:

We apply and contrast traditional and pipelined AES models to evaluate the efficacy of pipelined AES. Latency reduction, resource usage, and encryption speed are examples of performance measures.BothhardwareFPGAimplementationsandsimulationsbasedonPythonareused in the experiment. To quantify efficiency gains, data sets of different sizes are encrypted.

RESULTS AND CONSIDERATION:

TheexperimentalresultsshowasignificantreductioninencryptiontimeusingAES.

IMPROVEMENT OF ENCRYPTION SPEED:

The conventional process of the AES implementation process is consistent and leads to a higher delay.
AES Pipeline Demonstrates Reduced Execution Time Through Parallel Processing.
Hardware pipeline in FPGA implementations has shown speed improvements of up to 40%over non-pipelined implementations.

RESOURCE UTILIZATION:

FPGAimplementationsofpipelinedAESdemonstratedefficientusageoflogicgatesandmemory, leading to optimized performance.

Software-basedpipeliningshowedimprovedCPUcoreutilization,reducingidletimeandenhancing processing efficiency.

Reduced latency:

Traditional AES implementations perform sequential operations, which increases processing latency. Pipelining reduces latency by overlapping encryption rounds, which results in faster data encryption.

Difficulties encountered:

Pipeline dangers: Data dependencies between AES cycles can cause deadlocks and impact performance.

Hardware implementation complexity: Implementing a deep pipeline in an FPGA or ASIC requirescarefuloptimizationtoavoidexcessivepowerconsumption.Memoryoverhead:Software pipelining may require additional memory management techniques to efficiently handle parallel operations.

Results show that pipelining significantly improves the speed and efficiency of AES encryption, makingitaviableoptimizationforhigh-performancecryptographyapplications.However,careful optimization is required to manage pipeline hazards and equipment complexity.

REFERENCES:

1. Daemen,J.,&Rijmen,V.(2002).*TheDesignofRijndael:AES-TheAdvancedEncryption Standard*. Springer.

2. National Institute of Standards and Technology. (2001). *Advanced Encryption Standard (AES)*. NIST FIPS Publication 197.

3. Kocher, P. C. (1996). *Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems*. Advances in Cryptology—CRYPTO'96.

4. Chodowiec, P., & Gaj, K. (2003). *Very compact FPGA implementation of the AES algorithm*.In*InternationalWorkshoponCryptographicHardwareandEmbeddedSystems* (pp. 319-333).