Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 6 (2012) 872 – 879

Procedia Technology

2nd International Conference on Communication, Computing & Security [ICCCS-2012]

Inter Cipher Block Diffusion: A Novel Transformation for Proposed Parallel AES

Shashank Srivastava[a], Avinash Kumar Singh, G.C. Nandi

Indian Institute Of Information Technolgy, Allahabad, India

Abstract

With the advent of parallel computing, real time processing of large data encryption and decryption seems to be possible. Among many encryption standards, AES has gained popularity due to its high security with low acceptable cost. In real life applications, AES uses AES-CTR and ASE-EBC mode of operation to perform parallel encryption and decryption for large size of data, but it creates an opportunity for an adversary to find out the pattern at block level.

In this paper, we propose a novel transformation for parallel AES which leads to significant performance improvement with providing adequate security. To provide security with parallelism, a group of four input sub blocks of size 128 bits are processed simultaneously for creating a group of four output blocks of the same size. In order to wipe out the pattern at block level, we performed inter-block diffusion by adding a round key with all the four sub blocks in specified manner at the end of each round of parallel AES. In addition, complexity analysis proves that the proposed parallel AES approach is about 400 % faster than sequential AES.

"Keywords: Sequential AES, Parallel AES; Mode of Operation; Inter Block Diffusion"

1. Introduction

At the beginning of 21st century two Belgian researchers John Daemon and Vincent Rijment proposed a new algorithm of data encryption.. Their idea has later approved and publically announced by National Institute of Standards and Technology (NIST) in 2001 and James Nechvatal et al., in 2000. AES operations performs on byte rather than the bits makes this algorithm faster and the transformations used in the algorithm makes t secure.

Unlike DES, AES is not based on feistal cipher. AES is a modern block cipher technique that encrypts and decrypts data in chunks of 128 bits. It uses 10, 12 and 14 rounds according to the key sizes of 128, 192 or 256 bits respectively. In another words, AES encrypts each block of 128 bit by performing 10 rounds of encryption by different rounds' key which are generated by key expansion algorithm of AES.

In general practices, "Security of any encryption algorithm can be ensured until its cryptanalysis is not possible".

Corresponding author. Tel.: +91-9984905199
E-mail address: Shashank12march@gmail.com

doi:10.1016/j.protcy.2012.10.106

Due to its large key sizes, AES is more secure than DES. As DES uses 56 bit of key which is not resistant to brute force attack whereas AES uses 128, 192 and 256 bits of key for encipherment which dampens the brute force attacker. Here we have focused on those factors which make AES resistant to statistical, differential and linear analysis.

As we earlier stated, AES encrypts 128 bit block at a time and this encryption possesses N (10, 12, and 14) different round of operations with N different round's key. Each round of AES performs four transformations that are invertible in nature. The basic objectives of these transformations are to provide non linear relationship among plaintext, cipher text, key and to diffuse the plaintext's block at bit level.

AES performs substitution, permutation, mixing and round key addition at each round of block's encryption. Substitution involves 16 byte to byte transformation of 16 byte block (128 bits) can be called as intra-block transformation. Unlike DES, AES uses single S-Box for substitution which makes it efficient it terms of time complexity. After substitution of bytes, Next transformation permutes the bytes by shifting it row wise. (AES 128 bit block is represented in 16 byte i.e. 4*4 matrix of byte in which each cell of matrix denotes the hexadecimal representation of each byte) beautifully described by , U.S. Department of Commerce 2001, J. Daemen et al., in 2002 and 2010, as well as it is also well described in Forouzan and Stallings Books.

However substitution, transformation provides intra-block transformation, but in order to diffuse the plaintext block at bit level, the third transformation mixes bytes to achieve inter-byte transformation. Diffusion at bit level conceals the association between cipher text and the plaintext. Diffusion at bit level provides strength to AES against differential cryptanalysis attack which seeks the information about plaintext and cipher-text.

All above three transformations are invertible and the procedures of transformations are publicly exposed so it is easy for an adversary to find the relation between the plaintext and cipher-text. Therefore, at the end of each round AES uses the most important transformation based on round key, which is shared between only sender and receiver. AES uses key expansion algorithm to create N+1 rounds' key from a single cipher key.

In key expansion algorithm, each round's key is generated from its previous round key, so a chain is created among different round's key formulation. Since this dependency may leads to a pattern for cryptanalyst. So to make this dependency non linear, AES diffuses the bit pattern of each round's key by performing substitution and rotation operation.

1.1. Mode Of Operations And Security Issues

However, AES provides reliable security strength of transforming large size of data over the network, but somehow security strength is also dependent upon the mode of operation of encipherment. In real life, message or data may be of large or variable size i.e. much larger than the block size of encryption technique. In order to encrypt variable large size of data, five modes of operations have been devised as follows in Morris Dworkin in 2001.

- Electronic code book Mode(ECB)
- Cipher block chaining mode (CBC)
- Output feedback mode (OFB)
- Counter Mode (CTR mode)
- Cipher feedback mode (CFB mode)

Recently researchers proposed the concept of parallel AES implemented on CUDA (Computer unified devise architecture) to perform encryption in real time. It is mentioned in literature that electronic code book mode (ECB) can be used in parallel processing of encipherment described by Nhat-Phuong Tran et al., and Tomoiaga Radu Daniel et al., in 2011and by Di Biagio et al., in 2009.

However, authors performed parallelism on general purposed multi core processor architecture and on graphic processing unit using CTR mode but this parallelism generates pattern at block level which makes whole operation of encipherment vulnerable to differential cryptanalysis.

In ECB mode, each block of plaintext is independently encrypted; means encryption of different blocks can be performed simultaneously whereas in cipher block chaining mode or other feedback mode, each block encryption is dependent upon the previous cipher text block so parallel processing is not possible in these cases.

So, before delving into the details of our parallel AES, we give the brief introduction of electronic code book mode of encipherment. In ECB mode of operation, pattern at block level are preserved i.e. same blocks of plaintext generates same blocks of cipher text. Suppose in any case, if some blocks of cipher text are same, adversary will be able to know about the corresponding plain text block. The relation between cipher text and plaintext in ECB mode

can be understood by the relations C_i=E_k(P_i) and $P_i = D_k(C_i)$ means plaintext and cipher text is dependent on each other on the basis of key only. The block independency creates security loopholes in the security operations of parallel AES in ECB and CTR mode that would motivates the researchers who are working in the field of parallel AES.

Gaining the understanding of mode of operation and its security flaw, we proposed a novel transformation for each round of parallel AES to strengthen its security against cryptanalysis. The paper is constructed as follows. Section II describes the background details of sequential AES and gives the brief details of its cryptanalysis. In Section III, we proposed parallel AES with novel approach of inter-block diffusion transformation. Section IV performed complexity analysis of our parallel AES and show the comparison with sequential AES performance. In the last section V, we conclude the paper with its contribution towards the security and its future prospects.

2. Our Proposed Approach: Parallel AES

In order to speed up the process of the AES algorithm, we need the concept of parallelism. In general practices CTR and ECB mode of encipherment operation support parallelism discussed by Nhat-Phuong Tran et al., in 2011, but we earlier stated the security weakness of these mode of operations. So in order to achieve parallelism with security, we proposed our parallel AES algorithm based on novel inter-block diffusion concept. In modern block cipher approach, each encipherment technique works on the diffusion and confusion operation. DES provides bit by bit diffusion for 64 bit of block and for this it uses 8 S-box for each round.

AES provides bit diffusion by using single S- Box. AES works on byte rather than bit, but indirectly it diffuses the bits with the help of some complex transformations. In our parallel AES algorithm, we follow the same mode of operation like CTR and EBC to provide independent block of encipherment i.e. each block of encipherment is independent upon other blocks. However, AES –CTR mode provides intra-block diffusion but it does not talk about inter-block diffusion. Here we will see how we can achieve parallelism with inter-block diffusion without preserving the pattern at block level.

2.1. Inter-block diffusion:

Until now, we saw the operations performed on individual blocks independently, but in case of parallel AES, we will perform inter-block diffusion to discourage the cryptanalyst. In AES-CTR mode, pattern at the block level are preserved, is the main downside of the parallelism. We perform inter–block diffusion to disturb the pattern at block level in each round of AES encipherment.

2.2. Parallel AES:

Now, we see how parallel version of AES performs encryption. Here we only discuss about parallel AES for 128 bit.

Design Criteria:

In parallel AES, each round is same as the original design of AES. But here we assume the existence of multi-core processor. Each processor performs operation on each individual block independently in parallel manner. The main design components of parallel AES are as follows:

2.2.1. Data Unit:

Each data unit of parallel AES is of 64 byte block (4 sub blocks) and each sub block size is of 128 bits. Each sub block is processed independently in each round for three transformations, but the last transformation of each round, which is key dependent, performs inter-block diffusion. Each sub block of 128 bits is represented as 4*4 matrix. Parallel AES exploits seven measurement units: bits, bytes, words, sub blocks, state and block. Each block consists of four sub block of 128 bits .

2.2.2. Rounds:

Like sequential AES, parallel AES uses 10, 12 and 14 rounds depending upon the key sizes 128, 192 and 256 bits respectively. Each round possesses four transformations, from which three transformations works independently for each block while the last fourth transformation is key dependent transformation is performed on simultaneously on 4 blocks to provide inter-block diffusion.

As Figure 1 shows, N defines the number of rounds. It can be seen in figure for N round there should be N+ 1 round key. Here, four blocks of 128 bits are taken simultaneously in initial step and N rounds of encryption performed using N+ 1 key to create four sub cipher text blocks.

2.2.3. Structure of each round:

Figure 2 shows the construction of each round of parallel AES. Each round uses four transformations. Each transformation takes a state of 16 byte (4*4 Matrix) and performs operation on this state and changes it into a new output state which further becomes the new input state for the next transformation.

The working of each round of parallel AES is described by algorithm shown below. Initially input states $\boldsymbol{S_i}$ of four sub state of 128 bits (16 Bytes)$\boldsymbol{S_1, S_2, S_3, S_4}$ is taken for Substitution of bytes transformation. Sub Byte transformation changes the state $\boldsymbol{S_i}$ to state$\boldsymbol{S_B}$. The next ShiftRow transformation takes the state $\boldsymbol{S_B}$ as an input state and transforms this state it into $\boldsymbol{S_R}$ which is again acts as an input state for MixColumn transformation. MixColumn changes the state $\boldsymbol{S_R}$ to the state$\boldsymbol{S_M}$. Final inter byte transformation takes all four states of the state $\boldsymbol{S_M}$ and changes it to final output states $\boldsymbol{S_{IBD}}$. Output states consist of four 128 bits blocks. At the decryption site, the following inverse transformation is used: InvSubByte, InvShiftRows, InvMixColumns and InvInterBlockDiff.

2.2.4. Transformations:

To attain security, parallel AES uses four transformations at each round as follows:

- SubByte Transformation
- ShiftRow Transformation
- MixColumn Transformatin
- InterBlock Diffusion Transformation

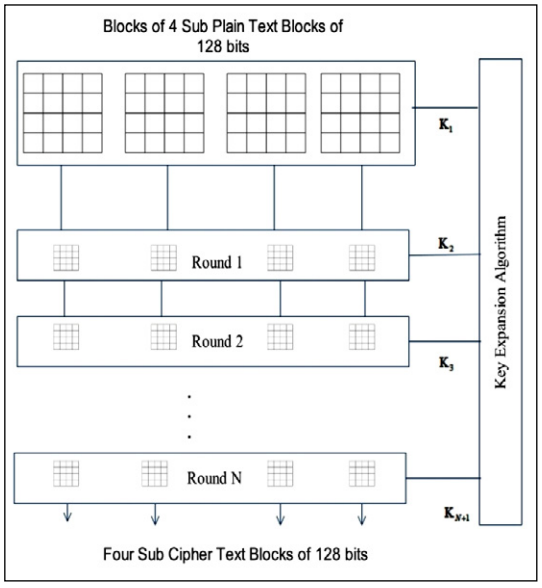

Figure 1. General Design of Parallel AES

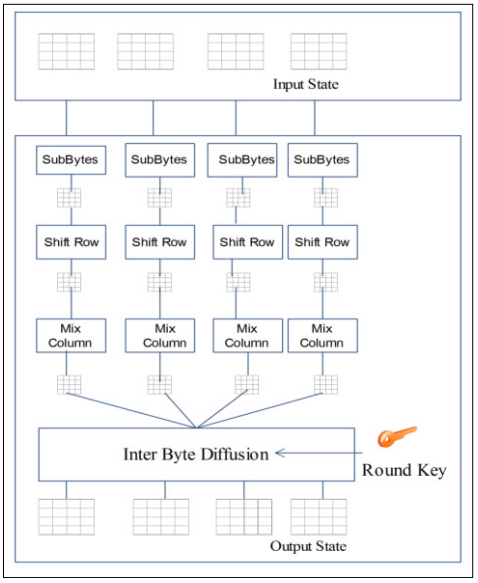

Figure2. Structure of each round at encryption site

The first three rounds of parallel AES is same as the sequential AES but it performed in parallel for four blocks at a time. InterByteDiffusion is the last transformation of each round. After mixed column transformation of four sub

states of 16 byte (4 * 4 matrixes) is created which acts as input states for InterBlock Diffusion transformation. This transformation diffuses the bytes of each state (Block) with others three blocks with adding round key.

Each state in parallel AES is represented as a combination of four sub states. Each sub state is represented by (4*4) matrix in which each cell represents 1 byte i.e. 2 hexadecimal digits. This transformation takes 4 sub states and diffuses them with the help of round key of 128 bits also represented as a 4x4 matrix. Due to this transformation, each block (state) is diffused with other three blocks which makes the cryptanalysis of parallel AES more complicated. At the decryption site, at each round, receiver has to first construct the original cipher blocks which have already been diffused amongst different blocks. (Refer notations from Table 3)

2.2.4.1. InterBlock Diffusion Transformation:

InterBlock diffusion transformation takes 4 blocks (4states after mix column transformation) at a time and diffused them according to algorithm 2. Table 2 describes the details of transformation.

2.2.4.2. Inverse Inter Block Diffusion transformation:

At the decryption site, inverse Inter Block diffusion takes placed. This transformation takes 4 blocks (4states after inter block diffusion transformation) at a time and construct outputs states as according to algorithm 3 illustrated in table 3.

Algorithm1: Activity performed in each round

Input States: S_i $i \in \{1, 2, 3, 4\}$ // State S_1, S_2, S_3, S_4
Output State: S_o // four output states
Transformation functions:
$SubByte( )$
$ShiftRow( )$
$MixColumn( )$
$IntBlockDiff( )$
$\forall\ i = 1{:}4$
$S_B, i \leftarrow SubByte(S_i)$
$S_R, i \leftarrow ShiftRow\ (S_B, i\ )$
$S_M, i \leftarrow MixColumn(S_R, i)$
$\forall\ i = 1{:}4$
$S_O \leftarrow IntBlockDiff(S_M, i, K)$ // Inter Block diffusion with add round key

Algorithm 2: Pseudo code for InterBlock diffusion Transformation

Input States: $S_{m,i}$ **// Each state is represented by 4 words of 4 byte each.**
Output State: $S_{ibd,j}$ $i: 0 \rightarrow 15$
Key: K_i $i: 0 \rightarrow 3$
Transformation function:
$S_{ibd,i} \leftarrow S_{m,j} \oplus K_{i\ mod\ 4}$
Initialization:
b= -1;
Begin:
for (i=0 to 15)
{
$q = i/4$;

$if((q + 5(i \bmod 4)) \bmod 4 == 0)$
$j \leftarrow q + 5((i-1) \bmod 4) + 1;$
$else$
$j \leftarrow q + 5(i \bmod 4);$
$S_{ibd,i} \leftarrow S_{m,j} \oplus K_{i \bmod 4}$;
if ($i \bmod 4 == 0$)
b++;
$S_b[\, i \bmod 4] = S_{ibd,i}$; // State of four words
}
End

Algorithm 3: Pseudo code for InverseInterBlock diffusion Transformation (At decryption side)

Input States: $S_{ibd,j}$ $i: 0 \rightarrow 15$ // $Words\ after\ the\ mix\ cloumn\ transformation$
Output State: $S_{m,i}$ $i: 0 \rightarrow 15$ // $Words\ after\ the\ InterBlockDiffusion\ transformation$
Key: K_c $c: 0 \rightarrow 3$
Transformation function:
$S_{m,i} \leftarrow S_{ibd,j} \oplus K_c$
Initialization:
$c = -1$
Begin:
$for(\, i = 0\ to\ 15)$
{
$k = i;$
if ($i \bmod 4 == 0$)
c++;
$if(\, c! = 0)$
$k = i - c;$
$j \leftarrow (k * 4) \bmod 16 + c;$
$S_{m,i} \leftarrow S_{ibd,j} \oplus K_c$;
$S_c[\, i \bmod 4] = S_{m,i}$; // State of four words

}**End**

Table 1.
Details of inter-Block diffusion

Output State1	Output State2	Output State3	Output State4
$S_{ibd,0} = S_{m,0} \oplus K_0$	$S_{ibd,4} = S_{m,1} \oplus K_0$	$S_{ibd,8} = S_{m,2} \oplus K_0$	$S_{ibd,12} = S_{m,3} \oplus K_0$
$S_{ibd,1} = S_{m,5} \oplus K_1$	$S_{ibd,5} = S_{m,6} \oplus K_1$	$S_{ibd,9} = S_{m,7} \oplus K_1$	$S_{ibd,13} = S_{m,4} \oplus K_1$
$S_{ibd,2} = S_{m,10} \oplus K_2$	$S_{ibd,6} = S_{m,11} \oplus K_2$	$S_{ibd,10} = S_{m,8} \oplus K_2$	$S_{ib,14} = S_{m,9} \oplus K_2$
$S_{ibd,3} = S_{m,15} \oplus K_3$	$S_{ibd,7} = S_{m,12} \oplus K_3$	$S_{ibd,11} = S_{m,13} \oplus K_3$	$S_{ibd,15} = S_{m,14} \oplus K_3$

Table 2.
Details of Inverse Inter-Block diffusion

Output State1	Output State2	Output State3	Output State4
$S_{m,0} = S_{ibd,0} \oplus K_0$	$S_{m,4} = S_{ibd,13} \oplus K_1$	$S_{m,8} = S_{ibd,10} \oplus K_2$	$S_{m,12} = S_{ibd,10} \oplus K_3$
$S_{m,1} = S_{ibd,4} \oplus K_0$	$S_{m,5} = S_{ibd,1} \oplus K_1$	$S_{m,9} = S_{ibd,14} \oplus K_2$	$S_{m,13} = S_{ibd,14} \oplus K_3$
$S_{m,2} = S_{m,8} \oplus K_0$	$S_{m,6} = S_{ibd,5} \oplus K_1$	$S_{m,10} = S_{ibd,2} \oplus K_2$	$S_{m,14} = S_{ibd,2} \oplus K_3$
$S_{m,3} = S_{m,12} \oplus K_0$	$S_{m,7} = S_{ibd,9} \oplus K_1$	$S_{m,11} = S_{ibd,6} \oplus K_3$	$S_{m,15} = S_{ibd,6} \oplus K_3$

3. Complexity Analysis:

In order to compare the efficiency of sequential AES with our parallel AES approach, we assume the mathematical representations which are shown in table.

3.1. Sequential AES:

S_{Block} =128 bits
$S_{Plaintext}$ = M*128 bits
$T_N = 9*T_R$.
T= T_N+T_{10}
$T_{P(seq)}$ = M*T
$T_R = T_{Sub} + T_{Shift} + T_{Mix} + T_{Key}$
$T_{10} = T_{Sub} + T_{Shift} + T_{Key}$
$T=9(T_{Sub} + T_{Shift} + T_{Mix} + T_{Key}) + T_{Sub} + T_{Shift} + T_{Key}$
$T = 10(T_{Sub} + T_{Shift} + T_{Key}) + 9*T_{Mix}$
$T_{P(seq)} = M\,(10(T_{Sub} + T_{Shift} + T_{Key}) + 9*T_{Mix})$

… (1)

3.2. Parallel AES:

S_B =4*128 bits
$M = \frac{S_P}{S_B} = \frac{M*128\ bits.}{4*128\ bits} = M/4.$
$T_R = T_{Sub} + T_{Shift} + T_{Mix} + T_{IBD}$
$T_{IBD} = 4*T_{Key}$
$T_{10} = T_{Sub} + T_{Shift} + T_{IBD}$
$T=9(T_{Sub} + T_{Shift} + T_{Mix} + T_{IBD}) + T_{Sub} + T_{Shift} + T_{IBD}$
$T= 9(T_{Sub} + T_{Shift} + T_{Mix} + 4*T_{Key}) + T_{Sub} + T_{Shift} + 4*T_{Key}$
$T= 10(T_{Sub} + T_{Shift} + 4*T_{Key}) + 9T_{Mix}$
$T_{P(par)}$ = M/4*T
$T_{P(par)} = = M/4*(10(T_{Sub} + T_{Shift} + 4*T_{Key}) + 9T_{Mix})$

… (2)

From equation (1) and (2) we conclude that

$$T_{P(par)} \approx 4\ times\ T_{P(seq)}\ .$$

4. Conclusions and future work

In this paper, we proposed a parallel AES algorithm with inter byte diffusion functionality. AES uses the CTR and EBC modes of operation in parallel processing of encipherment but it leads to generation of pattern at block level which may be exploited by an adversary. In parallel environment, to diffuse the blocks at each round, we proposed a novel approach of inter-block diffusion.

Inter-block diffusion wipe outs the pattern at block level by performing transformation at each round with adding round key. We presented parallel AES with inter-block diffusion transformation to reduce the computational complexity of algorithm with increasing the cryptanalysis complexity on the same pace. Doing complexity analysis, we found our approach about four times better than sequential approach. The future prospect of our algorithm is to diffuse the blocks at bit level.

References

James Nechvatal, Elaine Barker Lawrence Bassham, Morris Dworkin, James Foti, Edward Roback, 2000. ”Report on the development of the advanced encryption standard (AES)”, Technical Report published by NIST. http://csrc.nist.gov/archive/aes/round2/r2report.pdf.

Nhat-Phuong Tran, Myungho Lee, Sugwon Hong, Seung-Jae Lee, 2011. "Parallel Execution of AES-CTR Algorithm Using Extended Block Size," Computational Science and Engineering (CSE), IEEE 14th International Conference on , vol., no., pp.191-198.

Tomoiagă Radu Daniel; Stratulat Mircea, 2011. “AES Algorithm Adapted on GPU Using CUDA for Small Data and Large Data Volume Encryption”, in the proceeding of International Journal Of Applied Mathematics And Informatics, issue 2, Volume 5,pp. 71-81.

Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197 , U.S. Department of Commerce, National Institute of Standards and Technology Lab oratory (ITL), 2001 November 26.

J. Daemen, V. Rijmen, 2002. “The Design of Rijndael: AES The Advanced Encryption Standard”, Springer-Verlag.

J. Daemen, V. Rijmen, 2010. “AES Proposal Rijndael [EB OL]”, http://www.daimi.au.dk/~ivan/rijndael.pdf.

National Institute of Standards and Technology (NIST), “FIPS-197: Advanced Encryption Standard.” http://www.itl.nist.gov/fipspubs/ , Nov. 2001.

Morris Dworkin, 2001. “Recommendation for Block Cipher Modes of Operation”, book published in the Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology.

Behrouz A. Forouzan, Debdeep Mukhopadhay, “Cryptography and Network Security”, Second Edition, book published in Tata McGraw Hill press, ISBN: 0-07-070208-X, 2010.

William Stallings, “Cryptography and Network Security”, Fifth Edition, book published in Prentice Hall, ISBN: 0130914290, 2010.

Di Biagio, A.; Barenghi, A.; Agosta, G.; Pelosi, G.; 2009. "Design of a parallel AES for graphics hardware using the CUDA framework," *Parallel & Distributed Processing. IPDPS 2009. IEEE International Symposium on* , vol., no., pp.1-8, 23-29.

Table1.
Notations used in this paper

Notations	Meaning
S_B	States after substitution byte transformation
S_R	States after shift row transformation
S_M	States after mix column transformation
S_{IBD}	States after inter block transformation
$S_{m,i}$	i^{th}word of state S_M
K_i	i^{th} word of round key
$T_{P(seq)}$	Total execution time of sequential AES
T_R	Execution time of each round
$T_{P(seq)}$	Total execution time of sequential AES
T_R	Execution time of each round
T_N	Total execution time of all 9 rounds
$T_{P(par)}$	Total execution time of parallel AES
T_{Sub}	Time taken by substitution transformation
T_{Shift}	Time taken by shift row transformation