DESIGN AND ANALYSIS OF SECURE DATA ENHANCEMENT IN A DIGITAL MANUFACTURING SYSTEM

Giriraj Mannayee, C. Pandivelan, M. Boopathi and P Ponnusamy
School of Mechanical Engineering, Vellore Institute of Technology, Vellore, India

Vivekraj Mannayee
Deputy Manager, Faurecia Interior Systems, Chennai, Tamil Nadu., India

ABSTRACT

Over the recent years, information technology is transferred from the usage of mainframe computer systems to client-server models, embedding new principles, concepts and techniques. It is currently known as cloud computing. The primary motivation for embarking on cloud computing, as advocated by computer science experts, encourages users to abandon the burden of computer hardware needs, time – sharing utility demands and network system needs. It is also used to embrace the utility of commercial grid systems, which guarantee to reduce overall client – side requirements and system complexity. While being attracted to cloud computing, it is realized that there are several unchartered risks and challenges introduced along with this relocation decision. Primarily, effectiveness of the traditional protection mechanisms is deteriorated. Despite the above shortcomings of cloud computing, the utilization and attraction of cloud computing are still expanding inwardly from general to manufacturing – based cloud computing. Cloud computing aids in the creation of intelligent factory networks and prominently in crafting the features of product life- cycle services such as product design, manufacturing, testing and management. In this manufacturing- propelled cloud computing system, the traditional operating paradigm of shop-floor control system exists and it consists of material requirement planning (MRP), enterprise resource planning (ERP), master production scheduling (MPS), reorder point system (RPS), sales and operations planning (S&OP) has been abandoned. Instead, a coordinated system has all these planning and operation modules embedded into it. It has evolved directly and indirectly. This niche concept is referred as Manufacturing Execution System.

Keywords: Digital Manufacturing System, MES, Data Security, Cloud manufacturing


http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=9&IType=7
1. INTRODUCTION

Improve product tracking and visibility of stage operations, with the manufacturing execution system software. Integrated with enterprise business processes, Manufacturing Execution system creates the ideal data transfer for high-quality and efficient manufacturing system, [4,5] increased reliability, and full, global product traceability. [1,6] In order to achieve (i) Get real-time visibility into end-to-end manufacturing processes to support global coordination. (ii) Achieve complete traceability of products through their life cycles (iii) Optimize work in progress (WIP) in response to reduced lead times (iv) Improve quality and reduce variation through corrective actions that identify defective products (v) Boost plant performance and profitability (vi) Optimize productivity and use of resources and assets. But in this process we have faced one of the problems is intruders. In order to avoid intruder role we introduce in-between MES devices and ERP software encryption algorithm like AES, Serpent, Twofish, AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES, Twofish-Serpent. [12,13] Intermediate documents protected by above mentioned algorithm according to our functionality limit.

2. CLOUD MANUFACTURING EXECUTION SYSTEM EXPERIMENTAL SETUP

Figure 1 demonstrates cloud based assembling execution framework. In this framework is firmly incorporated with substantial numbers and sorts of Programmable Logic Control frameworks that drive gathering forms this level of coordination can't be successfully be obliged with current cloud advances. [2,11] Further, producing execution framework accessibility in the present condition of distributed computing would convey the gathering framework procedure to a stop.

![Figure 1 Cloud Manufacturing Execution system](image)

Most manufacturing organizations disconnect their industrial facility systems from their business arranges in order to not uncover their get together process control frameworks and manufacturing assets for the public clouds. [8,9,13] The MES system that focused on mechanical programming and hardware is a case of such hazard. We can imagine an on-preface MES interoperating with cloud-based applications. In fact, we think the truth for most assembling firms is that their endeavour IT engineering will be a half breed of on-commence and cloud-based applications. Intermediate documents are open from different gadgets like tablet, portable and social media packages..

2.1. Cloud based Manufacturing Execution Setup

Figure 2 demonstrates the exploratory setup for one of the cloud based get together framework. In this get together station comprises of different stages like item sorting, and gathering forms that can be controlled by programmable logic controller. The upper and the lower transports are driven by the turning actuator 1 and the lower transport revolving actuator 2 separately.
From the lot random selection choice of metallic pegs and plastic construct rings are put in light of the upper transport. The rings and pegs should be distinguished and isolated. This is finished by two nearness sensors, a closeness sensor1 and an infra-red intelligent sensor 2. By utilizing these two sensors a refinement can be made between the metallic and non-metallic pegs and the ring. By method for the sort straight actuator by 3, plastic rings can be shot out/dismisses down the get together chute, which can have up to plastic rings. Metallic segment, then, proceed on the upper transport and are absconded down the feeder chute. The feeder chute naturally sustains latches onto the lower transport. An infrared sensor is utilized to figure out if or not the gathering range is stacked state of exhaust. In the event that the condition is unfilled, the get together solenoid base rotational actuator is utilized to administer a ring from the get together channel into the gathering territory. The get together territory is situated quite recently over the lower transport and, when a metallic product passes, the latch draws in with the opening in the ring and the two parts are collected. The lower transport is utilized to convey finished parts into the accumulation plate. A printing module interfaced with the programmable logic controller, recovers information from the framework and prints over the segment with applicable parameters, similar to maker, permit no, bunch no, Mfg. Date, expiry date and so forth... In this work, a Siemens S7-300 programmable logic controller is utilized to control the procedure and programming called "Simatic Manager" is utilized to program the programmable logic controller.

2.2. Cloud Manufacturing Based Mes Data from a Programmable Logic Controller

On cloud based assembling floor have numerous MES based hardware. These MES apparatuses create transitional records. This middle of the road archives created with the assistance of programmable rationale controllers. Be that as it may, this cloud based MES information are effectively hacked or changed by interlopers. Encryption calculations additionally select element nature, in addition keeping in mind the end goal to expand open or private key verification too. The figures [3,4] indicates cloud based MES information.

![Figure 2](image)

**Figure 2** Manufacturing Execution system setup

![Figure 3](image)

**Figure 3** A report from a cloud based MES_01
3. SOURCE CODE FOR BLOWFISH ALGORITHM

Blowfish is a symmetric square figure that can be successfully utilized for encryption and shielding of information. It takes a variable-length key, from 32 bits to 448 bits, making it perfect for securing information. Blowfish was outlined in 1993 by Bruce Schneier as a quick, free other option to existing encryption calculations. Blowfish is unpatented and permit free, what's more, is accessible free for all employments. Blowfish Algorithm is a Feistel Network, emphasizing a basic encryption work 16 times. The piece size is 64 bits, and the key can be any length up to 448 bits. Despite the fact that there is a mind boggling instatement stage required before any encryption can occur, the genuine encryption of information is exceptionally effective on extensive microchips. Blowfish is a variable-length key square figure. It is appropriate for applications where the key does not change frequently, similar to a correspondences connect or a programmed record encryptor. It is fundamentally quicker than most encryption calculations when executed on 32-bit microchips with expansive information stores. Blowfish is a variable-length key, 64-bit piece figure. The calculation comprises of two parts: a key-extension part and an information encryption part. Key extension changes over a key of at most 448 bits into a few sub key exhibits totalling 4168 bytes. Information encryption happens by means of a 16-round Feistel organize. Each round comprises of a key dependent change, and a key-and information subordinate substitution. All operations are XORs and increases on 32-bit words. The main extra operations are four filed cluster information queries per round. The information change prepares the Blowfish Algorithm for encryption and Decryption, separately. The points of interest and working of the calculation are given below. Blowfish has 16 rounds.

The input is a 64-bit data element, x. Divide x into two 32-bit halves: xL, xR. Then, for i = 1 to 16:

\[
\begin{align*}
\text{xL} &= \text{xL} \text{ XOR } \text{Pi} \\
\text{xR} &= F(\text{xL}) \text{ XOR } \\
\text{xR} &\text{ Swap xL and xR}
\end{align*}
\]

After the sixteenth round, swap xL and xR again to undo the last swap. Then, xR = xR XOR P17 and xL = xL XOR P18. At last, recombine xL and xR to get the cipher text. Unscrambling is precisely the same as encryption, aside from that P1, P2, ..., P18 are utilized as a part of the turnaround request. Executions of Blowfish that require the quickest speeds ought to unroll the circle and guarantee that all sub keys are put away in reserve. /blowfish.cpp - written and placed in the public domain

/* Adapted for TrueCrypt */

#include <memory.h>  //include "Common/Tcdefs. h"  //include "Common/Endian. h"  //include "Blowfish. h"

http://www.iaeme.com/IJMET/index.asp 662 editor@iaeme.com
#define word32 unsigned __int32
#define byte unsigned __int8
#define GETBYTE (x, y) (unsigned int) (byte) ((x) >> (8* (y)))
#define ROUNDS 16
Static const unsigned __int32 p_init [16+2] =
{
  608135816U, 2242054355U, 320440878U, 57701188U, 2752067618U, 698298832U,
  137296536U, 3964562569U, 1160258022U, 953160567U, 3193202383U, 887688300U,
  3232508343U, 3380367581U, 1065670069U, 3041331479U, 2450970073U, 2306472731U
};
// this version is only used to make pbox and sbox
static void crypt_block(BF_KEY *key, const word32 in[2], word32 out[2])
{
  word32 left = in [0]; word32 right = in [1];
  Const word32 *const s=key->>box; const word32 *p=key->>pbox; unsigned i;
  left ^= p[0];
  for (i=0; i<ROUNDS/2; i++)
  {
    right ^= (((s[GETBYTE(left,3)] + s[256+GETBYTE(left,2)])
             s[2*256+GETBYTE(left,1)]) + s[3*256+GETBYTE(left,0)])
             p[2*i+1];
    left ^= (((s[GETBYTE(right,3)] + s[256+GETBYTE(right,2)])
           s[2*256+GETBYTE(right,1)]) + s[3*256+GETBYTE(right,0)])
           p[2*i+2];
  }
  right ^= p[ROUNDS+1]; out[0] = right;
  out[1] = left;
}
void BlowfishSetKey (BF_KEY *key, int keylength, unsigned char *key_string)
{
  unsigned i, j=0, k;
  word32 data, dspace[2] = {0, 1}; word32 *sbox = key->>sbox; word32 *pbox = key->>pbox;
  memcpy(pbox, p_init, sizeof(p_init)); memcpy(sbox, s_init, sizeof(s_init));
  // Xor key string into encryption key vector for (i=0 ; i<ROUNDS+2 ; ++i)
  {
    data = 0;
    for (k=0 ; k<4 ; ++k )
      data = (data << 8) | key_string[j++ % keylength]; pbox[i] ^= data;
  }
  crypt_block(key, dspace, pbox); for (i=0; i<ROUNDS; i++2)
  crypt_block(key, pbox+i, pbox+i+2); crypt_block(key, pbox+ROUNDS, sbox);
  for (i=0; i<4*256-2; i+=2)
  crypt_block(key, sbox+i, sbox+i+2); for (i=0; i < ROUNDS+2; i++)
Design and Analysis of Secure Data Enhancement in a Digital Manufacturing System

```c
key->pbox_dec[ROUNDS+1-i] = pbox[i];
}

void BlowfishEncryptLE (unsigned char *inBlock, unsigned char *outBlock, BF_KEY *key, int encrypt)
{
    word32 left = LE32 (((word32 *) inBlock)[0]); word32 right = LE32 (((word32 *) inBlock)[1]);
    const word32 *const s = key->sbox;
    const word32 * p = encrypt ? key->pbox : key->pbox_dec; unsigned i;
    left ^= p[0];
    for (i=0; i<RUNDS/2; i++)
    {
        right ^= (((s[GETBYTE(left,3)] + s[256+GETBYTE(left,2)])
                    s[2*256+GETBYTE(left,1)]) + s[3*256+GETBYTE(left,0)])
                   p[2*i+1];
        left ^= (((s[GETBYTE(right,3)] + s[256+GETBYTE(right,2)])
                   s[2*256+GETBYTE(right,1)]) + s[3*256+GETBYTE(right,0)])
                   p[2*i+2];
    }
    right ^= p[ROUNDS+1];
    ((word32 *) outBlock)[0] = LE32 (right); ((word32 *) outBlock)[1] = LE32 (left);
}

3.1. MES Data with Encrypted Format

The screen shot shows [5-7] different phases of encryption. Encryption implies that information is naturally scrambled or decoded just before is stacked or spared, with no client mediation. No MES information put away on a scrambled volume can be perused (unscrambled) without utilizing the right watchword/keyfile (s) or right encryption keys. [7,10,12] The whole document framework is encoded (e.g., record names, organizer names, substance of each record, free space, meta information, and so on).

![Encryption Options](image)

**Figure 5** Screenshot for MES data encrypt dynamic stage 1
3.2. MES Data With decrypt (mount) Format at End User

The screenshots [8-10] decrypt (mount) stages. Encrypted cloud based MES are like normal files. We provide the correct authentication of public key or private key and mount the original cloud based on MES intermediate file for the respective end user. [3,14,15] When we click the icon of the file The processor then begins loading a small initial portion of the file from the source of encrypted volume to RAM in order to view it. [6,12,14] While the portion is being loaded, crypt is automatically decrypting it. The decrypted portion of the file is viewed by the authenticated person. While this portion is being processed, the processor begins loading another small portion of the encrypted file from the MES intermediate-encrypted contents of RAM (memory) and the process repeats.
4. CONCLUSION

The "cloud MES as administration" model for cloud producing has risen with the ascent of the web advancements, in which; a specialist organization can give MES as a support of numerous customers over the Internet. Be that as it may, not at all like different administrations, as Saas, Pass, IaaS, databases is uncommon, in light of the fact that information is a valuable asset of a cloud based MES. In this paper results in a higher demand on information protection, security, honesty and accessibility to manufactures. To unravel
this, the "database as administration" model stores delicate information into the specialist organization in the wake of scrambling them, then confirmed individual alone decodes information. Besides in this paper contributed around a middle of the road record security framework with element nature. In prospect numerous car multinational organizations look forward towards cloud based manufacturing execution system. At that point naturally gate crashers part is significant. With a specific end goal to keep away from we present security frameworks. Yet, every one of the cases was not reasonable for security and confirmed framework. Wherever MNC’s is playing around transitional archive handling there we used encryption and unscrambling systems with element nature.

REFERENCES

Design and Analysis of Secure Data Enhancement in a Digital Manufacturing System


