한국정밀공학회지 제 24 권 제 5 호 (2007 년 5 월)
Journal of the Korean Society for Precision Engineering Vol. 24, No. 5, May 2007

CAA를 이용한 CATIA V5 파일보안시스템 개발에 관한 연구

채희창*, 박두섭#, 변재홍*

A Study on Development of CATIA V5 File Security System Using CAA

Hee Chang Chae*, Doo Seob Park# and Jae Hong Byun*

ABSTRACT

CATIA V5 is one of the most preferred softwares in product design for domestic and industrial use. But with the development of the IT industry, design data by CATIA V5 can easily be hacked and stolen especially via the internet and through assistance storage medium. The design data could be protected through executive, physical and technical security system. the best way to maintain confidentiality of data from unauthorized access is to have a cryptosystem of the technical security. In this paper, a cryptosystem for the protection of design data was being proposed.

The memory contains the file information made by the New and Open function of CATIA V5. No error can be expected even if the file changed before of after the application of Save and Open function. A cryptosystem was constructed in CATIA V5 by inserting crypto algorithm before and after the I/O process. The encryption/decryption algorithm of each function was based on the complex cipher, which applied permutation cipher and transpose cipher. The file security system was programmed in CAA V5 and Visual C++.

Key Words : CAA V5, Encryption (암호), Decryption (복호)

1. 서론

자동차 또는 항공기 업체와 같은 대형 제조업체들에서부터 중견기업에 이르기까지, 거의 모든 산업부문에서 생산되는 제품들이 CATIA를 통해 설계 및 제조되고 있다. 국외는 Boeing, Chrysler, BMW, Sony, Toyota 등과 국내는 현대/기아, 대우, 삼성, 삼립산업, 만도, 금호/한국 타이어 등 국내 자동차산업 및 항공 산업에서 CATIA 시스템이 설치되어 활용 중이다. 그렇지만 Window/Unix를 바탕으로 CATIA V5가 설치된 수많은 컴퓨터들이 인터넷으로 상호 연결되어 정보의 공유가 가능해짐에 따라 외부인의 무단 자료 유출 가능성은 항상 내재되어 있으며 자료 보관을 목적으로 하는 보조기억매

☞ 접수일: 2006년 10월 2일; 게재승인일: 2007년 2월 27일
* 전북대학교 기계설계학과
# 교신저자: 전북대학교 기계설계학과
E-mail: chaeprof@chonbuk.ac.kr Tel. (063) 270-2455

체에 의한 자료 유출 가능성은 더욱 가중되고 있다. 따라서 CATIA V5가 설치된 PC에서의 보안을 위해 대형 시스템과 같이 종합적이고 체계적인 대책이 필요하다. 그러나 PC의 경우 그 특성상 전문 오퍼레이터의 고용, 별도의 안전한 장소 확보 등 대형 시스템과 같은 수준의 보안체계 유지는 곤란하다. 그러므로 PC가 공유하는 환경에서 이에 적절한 체계적인 보안 대책이 다음과 같이 요구된다.[9,10]

첫째, 환경적, 행정적 보안으로써 PC 노출환경에 따른 시스템의 고장 등을 고려한 규율적인 대책과 보안의식 고취 및 시스템의 사용절차를 규정하는 등의 보안 절차를 말한다. 둘째, 물리적인 보안으로써 외부적인 보호대책을 위한 인원통제, 출입문의 견고한 잠금장치 등을 들 수 있다. 마지막으로 기술적인 보안 대책으로써 시스템 내부로의 접근을 통제하기 위한 락(Lock)기능이나 인증기능, 그리고 불법적인 자료접근이나 우연한 노출로부터 자료의 비밀성을 유지하기 위한 자료의 암호화 방법 등이 있다. 현재의 컴퓨터 내의 CATIA V5 파일은 제도적, 행정적인 보호대책과 물리적인 보호대책을 사용하여 보호되고 있지만 혹시라도 자료가 유출되었을 경우 CATIA V5 또는 뷰어(Viewer)가 설치된 어느 장치에서든지 열고 정보를 빼낼 수 있기 때문에 이는 완벽한 대책이 아님에는 틀림이 없다. 이에 우연한 노출로부터 자료의 비밀성을 보장하는 가장 좋은 방법은 CATIA V5를 위한 암호시스템을 갖추는 것이라 하겠다.[13]

본 연구에서는 CATIA V5 파일의 최종 보호수단은 데이터 자체를 암호화하여 보관하는 것이라는 점에 착안, 소프트웨어 CAA(Component Application Architecture) V5와 Visual C++를 사용하여 암호시스템을 CATIA V5 자체에 구현함으로써 사용자가 암호시스템에 신경 쓸 필요 없이 원하는 데이터를 암복호화 되게 하는 파일 보안 시스템 개발하고자 한다.

2. 시스템의 기본 구성

2.1 암호알고리즘

CAA V5는 CATIA V5를 사용자정의(Customize)할 수 있음을 이용하여 임의의 암호시스템을 CATIA V5 자체에 포함시킴으로써 개발에 사용된 프로그램의 소스가 공개되지 않는 한 사용된 암호가 어떤 종류의 것인지 모르게 할 수 있다. 이는 사용된 암호를 고의로 노출하지 않는다면 비밀성은 유지되므로, 파일을 암호화 하는데 가장 효율적인 방법은 암호화 키와 복호화 키가 같은 대칭 암호화 방식이라 할 수 있다. 이에 본 논문에서는 파일보안 개발에 대칭 암호화의 종류인 치환암호(Permutation cipher)와 전치암호(Transpose cipher)를 응용한 암호를 사용하였다.

2.1.1 치환암호

주어진 문자들의 열을 D개씩 나누어 각각의 위치를 치환시킴으로써 만들어지는 암호를 치환암호라 한다. 집합 I_n={1, 2, ⋯, n}에서 I_n으로의 전단사 함수를 집합 I_n위에서의 치환이라 한다. 이 때 치환

$$1 \rightarrow i_1,\ 2 \rightarrow i_2,\ \cdots,\ n \rightarrow i_n$$

을 $\begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}$으로 나타낸다.[2,3]

2.1.2 전치암호

기원전 400년경 희랍인에 의하여 폭이 좁은 긴 띠 모양의 종이를 곤봉에 감고 가로로 평문을 쓴 후에 종이를 풀어놓으므로 문자가 재배열 되도록 하는 Scytale 암호 방법을 이용하였다. 이는 인류의 역사상 가장 오래된 암호로 알려진 전치암호의 대표적인 예라 할 수 있다.

전단암호는 주어진 평문을 문자들의 수의 약수인 i에 대하여 한 행에 j 개의 문자들을 나열하여 i 개의 행, 즉 $i_n \times j_n$행렬을 얻고, 이의 전치행렬 $j_n \times i_n$행렬을 만들어 다시 나열하여 암호화한다.[4]

$$\begin{pmatrix} i_1j_1 & i_1j_2 & \cdots & i_1j_n \\ i_2j_1 & i_2j_2 & \cdots & i_2j_n \\ \vdots & \vdots & \ddots & \vdots \\ i_nj_1 & i_nj_2 & \cdots & i_nj_n \end{pmatrix} \rightarrow \begin{pmatrix} j_1i_1 & i_2j_1 & \cdots & i_nj_1 \\ i_1j_2 & i_2j_2 & \cdots & i_nj_2 \\ \vdots & \vdots & \ddots & \vdots \\ i_1j_n & i_2j_n & \cdots & i_nj_n \end{pmatrix}$$

2.1.3 사용된 암호

CATIA V5 암호시스템을 구축하는데 사용된 암호는 치환암호와 전치암호를 응용하여 새로운 암호를 만들어 사용하였다.

i) 암호화 시킬 평문을 10개열로 배열하고 임의의 문자 10개로 만들어진 행을 마지막에 더한다.

즉 평문에 의해 만들어진 $i_n \times j_n$ 인 행렬의 마지막에 1개의 행을 삽입하여 $i_{n+1} \times j_n$행렬을 만든다.

ii) 각각의 열을 주어진 치환함수 f 에 맞게 재배열한다.

iii) ii)에서 얻은 행렬을 전치시킨다.

즉, 사용된 암호는 ii)에서 치환암호의 원리를 이용하였으며, iii)에서 전치암호의 원리를 이용하였다.[2,4]

치환암호의 함수

$$f = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 2 & 6 & 9 & 8 & 7 & 1 & 0 & 3 & 5 \end{pmatrix}$$

삽입할 문자 행

[a b c d e f g h I j]

2.2 CAA V5

CAA V5는 CATIA, DELMIA, Deneb, CATweb 및 ENOVIA와 같은 다쏘시스템(Dassault Systems)사에서 개발된 모든 제품 라인에 대한 공통적인 애플리케이션 아키텍처이다. 전통적인 API들이 코딩스텝만을 나타내는 반면, CAA V5는 High-Level API로써 소프트웨어 개발 프로세서 전반을 나타내며, 개발을 위한 초기 엔지니어링부터 유지보수 및 업그레이드에 이르는 소프트웨어 제품 수명 주기 전반을 포괄하고 있다. 또한 CAA V5는 자신들의 기술을 V5에 접목시키고자 하는 3rd-Party 벤더들을 위한 새로운 파트너쉽 구조라고 할 수 있으며 여러 부문에 관련된 개발팀이 함께 일할 수 있도록 하여 소프트웨어 개발 워트플로우 관리와 관련 팀워크를 도와 전체 개발 과정을 가속화시켜 최적화한다.[8,12]

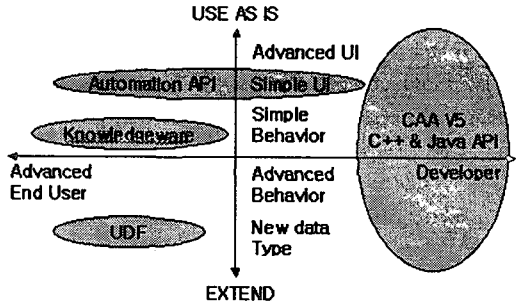

Fig. 1 CAA V5 Positioning

CAA V5는 Fig. 1과 같이 그 입지를 지정할 수 있다.

3. 파일보안시스템의 구현

3.1 보안시스템의 설계

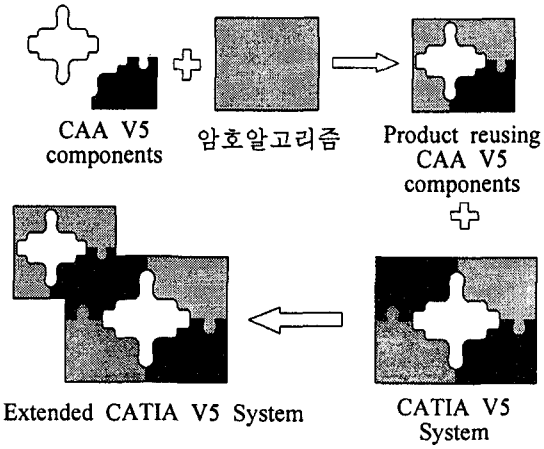

Fig. 2 Security system construct

CAA V5에서 암호시스템이 CATIA V5에 삽입될 수 있는데 필요한 Components를 호출하여 원래의(Original) CATIA V5 시스템에는 없는 새로운 기능(Security Open, Security Save, Security Save As, Security Save All)을 추가하면 확장된(Extended) CATIA V5시스템이 된다. 새로운 기능을 추가하기 위해서는 CATIA V5에서 Save와 Open할 때 파일에 관한 정보가 관리되는 위치와 호출되는 함수를 알아야 한다. 이는 Save와 Open의 이벤트를 실행할 때 적절한 CAA V5의 Components를 호출하여 암호알고리즘을 적용해야 하기 때문이다.

3.2 암호시스템의 설계

CATIA V5에서는 New에 의해 생성된 파일에 관한 정보를 메모리에서 관리를 한다. 또한 기존에 생성되어진 CATIA V5 파일을 Open할 경우도 메모리로 Open한 후 실제 파일로 작업하는 것이 아니라 파일과의 연결을 끊어 메모리 상의 파일로 작업을 하게 된다. 이는 CATIA V5에서 사용하고 있는 모든 파일의 정보는 메모리 상에서 존재한다는 말이 되고, 만약 Save와 Open하기 전과 후에 파일(기존 파일, 새로 만든 파일)을 조작하여도 오류가 나지 않는다는 것을 알 수 있다. 암호시스템은 이 특성

을 이용하여 Save와 Open의 이벤트 전후에 파일의 임의의 조작을 하여 암복호화 된다.

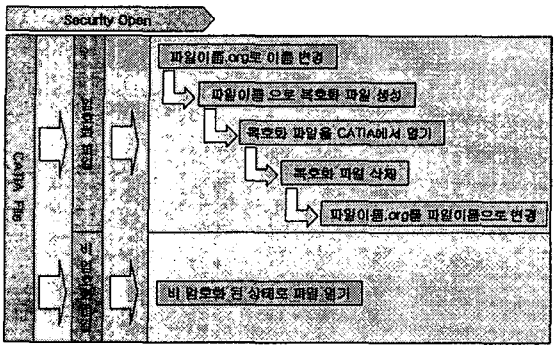

Fig. 3 Algorithm of security open

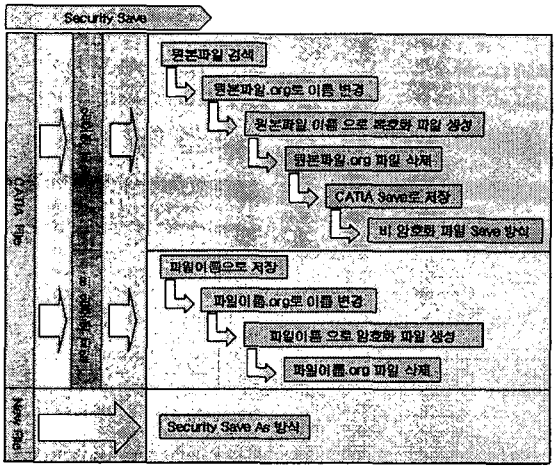

Fig. 4 Algorithm of security save

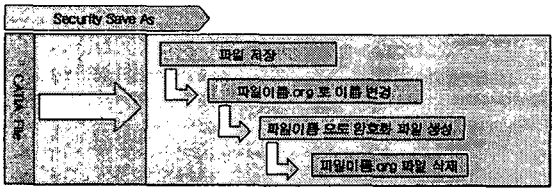

Fig. 5 Algorithm of security save as

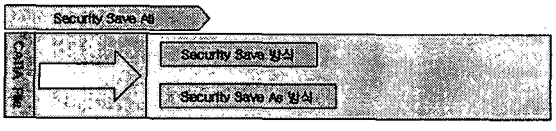

Fig. 6 Algorithm of security save all

4. 보안시스템의 검증

CATIA V5에서 저장/생성되는 파일형식은 바이너리(Binary)이지만 검증하기 쉽게 하기 위하여 본 논문의 암호시스템 정상작동 유무를 확인하는 방법으로 데이터파일의 바이너리 내용을 Hexa값과 ASCII 코드 값으로 표현할 수 있는 유틸리티 IDM Software Solutions의 UltraEdit-32를 이용하였다.

Fig. 7과 Fig. 8은 바이너리 형식으로 저장된 CATIA V5 파일을 Hexa값과 ASCII 코드 값으로 나타낸 것이다. 그 중에 Header부분의 일부인 0~99(100)번째를 파일암호화에 사용되었다.[8]

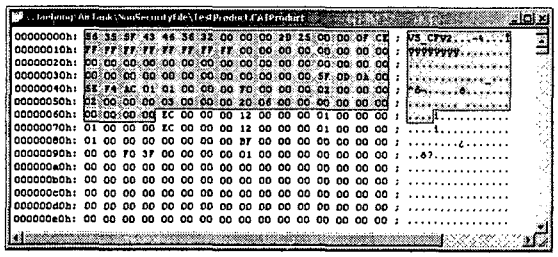

Fig. 7 Original File

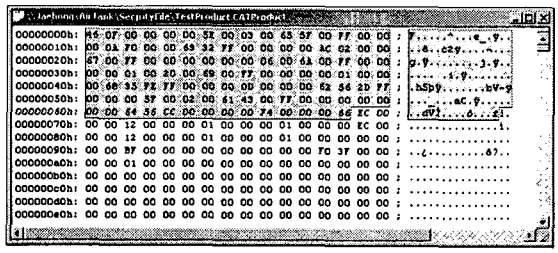

Fig 8 Security File

Fig. 9와 Fig. 10은 CATIA V5 파일의 임의의 부분을 파일암호화 하는데 사용이 가능함을 보여주기 위해 1999~2099(100개)번째를 사용해 파일을 암호화 시켰다.

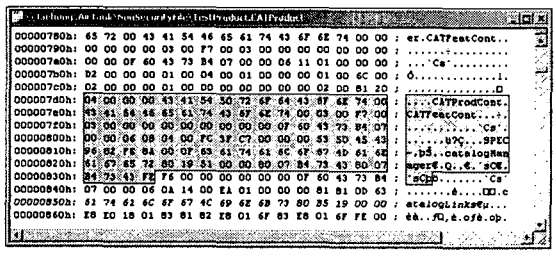

Fig. 9 Original File

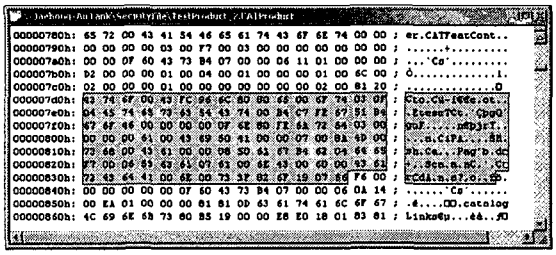

Fig. 10 Security File

본 논문의 암호시스템의 코드를 조금만 수정하는 것으로 여러 부분을 한꺼번에 암호화하기, 삽입된 문자열을 변환, 암호알고리즘의 변경 등을 할 수 있어 개발자가 원한다면 추후에 더욱더 강력한 보안시스템으로 얼마든지 변경이 가능하다.

5. 결론

본 연구는 현재 산업 현장에서 IT분야의 급속한 발달로 인하여 모든 산업의 데이터를 컴퓨터를 이용하여 처리, 관리, 전달하는 상황에서 그로 인해 발생되는 CATIA V5의 데이터파일 유출 및 도난 등의 예상 위험에서 데이터의 최종보호 수단은 파일자체를 암호화하여 보관하는 것이라는 점에 착안하여 사용자가 CATIA V5에서 임의의 파일을 로컬 디렉토리(Local Directory) 또는 파일서버(File Server)에 저장과 열기할 때 자동으로 암복호화 되도록 하는 파일보안시스템을 개발하는 연구를 수행하였다.[7] 암호시스템 개발에는 CATIA V5의 개발 API인 CAA V5, Visual C++과 암호알고리즘을 사용하였고 CATIA V5 자체에 개발된 보안기능을 편리하게 사용할 수 있도록 각 커맨드(Command)와 링크된 아이콘들을 포함한 툴바를 삽입하였다.

차후 연구로는, 현재 보안시스템은 관리자형으로써 보안기능 뿐만 아니라 기존의 CATIA V5에 있는 저장과 열기 기능이 아직도 메뉴의 풀다운(Pull-Down)과 기존툴바 안에도 존재하고 있어 사용자가 파일암호의 유무를 결정할 수 있다. 따라서 CATIA V5를 원래에 있는 저장과 열기 기능을 없애거나 기존기능을 보안기능으로 대체를 시켜 파일유출 가능성이 없는 사용자형을 만들어 향후 파일보안시스템을 관리자형과 사용자형으로 구분하는 연구를 수행하고자 한다. 또한 보안 알고리즘 역시 전체 파일 암호화와 속도 향상을 목표로 연구를 수행하고자 한다.

참고문헌

1. Douglas, R., "Crytography-Theory and Practice," CRC Press Inc., 1995.
2. Alfres, J., Paul, C. and Scott, A., "Handbook of applied Cryptography," CRC Press Inc., 1997.
3. Koblitz, N., "Algebraic aspects of cryptography," Kluwer Academic Publishers, 2000.
4. Stallings, W., "Network and Internetwork Security," IEEE Press, 1995.
5. Advanced Computer Aided Design User's Manual
6. Kruglinski, D. J., "Inside Visual C++ Fourth Edition," WP Publishers & Distributors, 1999.
7. Richter, J., "Programming Application for Microsoft Windows," Fourth Edition, Microsoft Press, 1999.
8. Lee, L.Y. and Song, Y.Z., "Modern Cryptography," Life & Power Publishing Co.,Ltd., 1997.
9. Park, C. S., "Cryptography-Theory and Security," Daeyoungsa, 1999.
10. Lee, M. S., "Modern Cryptography," KYOWOO Publishing Co.,Ltd., 2001.
11. Won, D. H., "Modern Cryptography," Green Publishing Co.,Ltd., 2003.
12. Lee, K. W., Lee, J. H., Kim, J. H. and Oh, C. S., "Design and Analysis of Data File Protection based on the Stream Cipher," The Korea Contents Association, Vol. 4, No. 1, pp. 56-66, 2004.
13. Kim, H. J., Lee, S. Y. and Chu, Y. Y., "File Security Systems in PDA," Korea Multimedia Society, pp. 91-94, 2003.