19

Authentication of product data and product components

R. Anderl; M. Momberg
Fachgebiet Datenverarbeitung in der Konstruktion
Darmstadt University of Technology
Petersenstrasse 30, D-64287 Darmstadt
Tel. +49 (0)6151/16-6001, Fax +49 (0)6151/16-6854
{anderl,momberg}@dik.maschinenbau.tu-darmstadt.de

Abstract

With the increasing use of product data technology and product models in particular in the production phase of the product life cycle, approval mechanisms, often integrated in workflow models, show shortcomings when they are used as an electronical successor of the well known signature below a technical drawing. When considering the aspect of the creation of a legal document with an approval, questions may arise about the guarantee and provability of the approval authorship and integrity of the approved partial model data because of the ease of modifying data in computers.

The paper describes how cryptographic methods can be integrated in the approval process to provide a way for establishing the authenticity of the approval concerning both the authorship and the integrity.

When leaving the production phase and advancing to the use and, in particular, the recycling phase, it appears crucial that the relationship between approved product data used for the manufacturing of a part or assembly and the part or assembly itself is only unidirectional. In case of liability, a secure, i.e. unforgeable, method is needed to map from the part to the product data from which it was manufactured and especially to the approvals which accompanied the production of the part. Similarly, a recycler might be interested in the original composition of an assembly concerning the materials used in it, e.g. lubrication liquids.

The paper discusses established as well as new methods to tag parts and assemblies with the objective to obtain a similar level of authenticity for the part as for the product data which described it. The integration of the tagging methods in the production phase will be treated as well.

J. J. Mills et al. (eds.), *Information Infrastructure Systems for Manufacturing II*

Keywords
Product data technology, Cryptography, Digital signature, Message digest, Hash-Function, Steganography, STEP

1 INTRODUCTION

Product data is the generic term for all those data that describe a product, i. e. all assemblies and parts, as complete as possible under the given circumstances. Often this term also includes the data necessary for manufacturing the product, for instance tool descriptions, and for managing the development workflow like approvals, work orders, or activities. Separated from the product data, the so-called meta data is managed which is needed for describing the product data structures.

When the design of a product component, an assembly or part, reaches a certain degree of maturity, it is subject to an approval by one or more responsible individuals. The classical approach for an approval is to provide a technical drawing of the component which is then examined by the approval engineer(s). The drawing is either rejected and returned to the designer for revision or accepted and thereby approved to be used as a basis for further development or for manufacturing. The drawing, equipped with the approver's signature, is forwarded to the product documentation department. It now constitutes a legally binding document for product liability purposes.

Although the integration of computer systems in the product development process has proceeded to a state where component geometry is solely created utilizing three-dimensional computer aided design systems (3D-CAD systems), the approval process yet remains conservative. A two-dimensional technical drawing is derived from the 3D model of the component and plotted on paper. The paper drawing is examined, approved with signature, and archived, or, in case of a rejection, sent to the designer who changes the marked inconsistencies in the 3D model and initiates another approval. Instead of drawing libraries or microfiches today electronical methods are often used for archiving. The drawing is scanned, converted to a graphics file format that is likely to be readable for part or whole of the archiving time provided by law, and then archived on a storage media, mostly of the optical write once read many (WORM) type.

As it is desirable to avoid this type of media discontinuity coupled with the incompleteness of the archived product data, an approval of the 3D model seems to be logical. The hurdle to be taken is the proof of authenticity of the product data necessary for the acceptance as a legally binding document.

One way to take the hurdle is to use digital signatures as a surrogate for manual signatures. The first part of this paper explains what digital signatures are and how they may be used in the digital approval process.

After the product components have been manufactured, they are assembled and the resulting product is offered. In the following utilization phase the product needs maintenance or, in case of a failure, a repair. The product may even have

caused an accident which leads to legal proceedings. And after the utilization phase, when the product has become worn or is no longer able to fulfill its function, it is disposed or parts of it will be recycled. In all mentioned cases product data must be retrieved by information that had been attached to the product component by some method during the manufacturing phase. Furthermore the producer may need to prove that all parts of the product in question are genuine or rather that some parts are not. In the second part of the paper the structure of the tagging information is detailed and, for each identified layer, already established and new transformation methods are discussed, the new ones with respect to obtaining the authenticity of the tagging information or the component itself.

2 PRODUCT DATA AUTHENTICATION

The product data of a particular product, as it is stored in product data management (PDM) systems, can be divided into partial models describing certain views on assemblies or parts, for instance, a geometry view or a kinematic analysis view, and associated process-oriented data, e. g. an approval of a component design or a work order to manufacture a part or to further develop dependent components. A record for the latter type of data usually contains the information who initiated the activity, when it was initiated, who authorized it, what was to do, when was it to be accomplished, and who has done it. The semantics of an approval differ slightly but contain the same relationships to the parties involved in the approval. The record is set up from the partial model that has been approved, the approval date, the approval status (for example, approved for stability calculations, approved for manufacturing, or approved for resource planning), and the approvers.

A forger who is able to gain sufficient access to the PDM system may later on change the approved partial model or other data of interest without leaving a noticeable hint about this illegal change. Sufficient access means that he may be able to break any conventional security method like marking the data as read-only, separating the involved computer systems from a larger network, or keeping the computer systems locked in a suitably secured room.

Digital signatures

In the emerging electronic business over the Internet a method known as digital signature is used to cope with this kind of problem. A digital signature is created by the signer's application of cryptographic algorithms on the document to be signed. An individual may verify the digital signature by applying corresponding cryptographic algorithms. A successful verification not only states that the document was actually signed by the signer but also that the document was not changed afterwards which is just the set of features demanded from the above example.

The subject of the cryptography is to keep messages secret. As this can only be accomplished if breaking of the secrecy is tried and its cost may then be estimated, the sibling science of cryptanalysis was founded. Both constitute the science of cryptology (Schneier (1996)).

Cryptography suggests to encrypt the message with an encryption algorithm using a secret piece of information, a so-called key, which is only known to a limited circle of individuals. The encrypted message, the cipher, can only be transformed to the original message if the corresponding decryption algorithm and another or the same key that was used for the encryption are known. The knowledge of the encryption and decryption algorithms will not give any advantage to a potential attacker if the algorithm pair is sufficiently secure, so they are usually published. The only secret is the key or the keypair which consist of a bit sequence of finite length l (currently l's between 128 and 1024 Bit are used). Because of the key's finite length all possible bit combinations may be tested in the decryption of an interesting cipher, needing 2^{l-1} tests in the mean to find the key. Therefore, this type of cryptanalytic attack is called brute force attack. The only measure to fight this attack is to choose longer keys and algorithms which may use them. As each test takes a certain amount of time depending on the computing power available to the attacker, increasing the key length by one doubles the time for a successful attack. If this time exceeds the lifetime of the encrypted data, the keys and corresponding algorithms are rated as cryptographically secure.

Public key algorithms

The concept for digital signatures that is mostly used these days was invented by Whitfield Diffie and Martin Hellman in 1976 (Diffie and Hellman (1976)). It is based on so-called public key algorithms which use keypairs instead of single keys for both encryption and decryption. If a message is encrypted with the first key, it can only be decrypted with the second key (and vice versa). The idea was to keep one key secret and publish the other key to all interested individuals. Then the owner of the secret key may encrypt a message and send the cipher to someone owning the public key.

When the recipient is able to decrypt the cipher with the public key, he knows that the message was encrypted by the owner of the secret, private key. This way the cipher acts as a signature of the message which may later be verified by applying the public key.

Certificates

The keypair for a public key algorithm is created by the later owner of the private key but a potential recipient of an encrypted message signed by the private key owner needs the public key in order to verify the signature. To provide this key in a secure way so that no one can change it on the way to the signature verifier is a challenging problem which is solved today by the foundation of so-

called certification authorities. The creator of the keypair meets with a certification authority, proves his identity by showing a passport or some other identification, and passes his public key. The certification authority returns its own public key for later usage. A potential message recipient will do the same, thus owning the public key of the certification authority too.

If the recipient actually receives a signed message but not yet owns the suitable public key, he may request the key from the certification authority which in turn sends the key encrypted with its private key. As the recipient owns the public key of the certification authority, he may decrypt this message, thereby verifying the signature of the certification authority. If the recipient trusts in the certification authority's integrity, he assumes the just decrypted public key to be authentic. Therefore, he may try to decrypt the received message and, if successful, accept the message as being signed by the private key owner.

Digital fingerprints

Public key algorithms tend to slow down significantly when the message that is to be signed is rather long (i. e. the time complexity of those algorithms with a given message length is polynomial with a polynom degree $d > 2$ or even exponential). To allow a long message to be signed in an acceptable time, a so-called one-way hash-function is applied to the message yielding a fixed-length bit sequence, the digital fingerprint or message digest, which identifies the message. The hash-function should hold the prerequisites that from a given fingerprint the original message should not be computable, from a given fingerprint a second message with the same fingerprint should not be constructable, and the hash-function should be fairly easy to evaluate.

Signing process-related product data

The application of digital signatures on process-related product data like the above mentioned approval or work order is straight forward and is explained in the following by choosing the approval as an example.

Each approver should create a keypair, keep the secret key by his own (in particular not integrate it in the product data), and deposit the public key in the product data record describing his personality, preferably accompanied by a certificate of a certification authority. To simplify the task of public key management, only one certification authority should be involved in a product development and the authority should be trusted by all participants of the development process. Furthermore, the certification authority should be expected to survive the lifetime of the product data including the archiving time to guarantee the provability of the authenticity of all public keys used in the development process for the whole time period.

When an approver has approved a partial model of the product data, he signs the approval by collecting and ordering all data necessary to describe the approval including the approved partial model, computing the digital fingerprint of it,

encrypting the fingerprint with his private key, and attaching the signature to the approval representation in the product data.

An individual who wants to verify the signature needs to collect and order all the data the same way the approver did, compute the fingerprint again, fetch the approver's public key from the product data with possibly verifying it with the accompanied certificate, decrypt the signature with the public key, and compare both fingerprints for equality. If the signature cannot be decrypted, either the public key is wrong and thus the certificate is not trustworthy, or the signature was not performed by the approver. If the fingerprints aren't equal, the product data has been changed, given the collecting and ordering algorithm were the same for both approval signing and signature verification. In the last case, the fingerprints are equal, proving the authenticity of the approval.

Discussion of the signing process

Some points of interest should be mentioned about the above approval signing process. The approver improves the quality of his signature if he includes the signing date and time in the fingerprint computation. This way the approval is fixed to the one date which cannot be changed afterwards without notice.

The list of product data that is collected for building the fingerprint also determines the quality of the signature because data elements that are not included in the fingerprint will not be signed and may therefore be subject to unverifiable changes by forgers. The ordering of the data collection must be identical for both the signing and the verification process because of the nature of the hash-function which performs a linearization of the graph-like product data structure.

The algorithm for computing the digital fingerprint turns out to be the central component of the signing process because it has to traverse the whole data collection with all dependent structures in a predefined and repeatable manner. Dobbertin (1996) states thatafter the public key encryption/decryption algorithms were subject to extensive attacks, the interest will turn to the cryptographic security of the hash-functions in the near future. Thus, designing hash-functions from scratch is very dangerous from a cryptanalytical point of view. In consequence, the utilization of well known and thoroughly analyzed algorithms is recommendable. The main disadvantage of those algorithms, from whom MD5 (Rivest (1992)), SHA (NIST (1993)), and RIPEMD-160 (RIPE Consortium (1992)) are the ones most widely used, in the context of product data fingerprinting is their presumption of a linear byte stream as input. Product data structures usually constitute an unconnected graph, i. e. a set of connected graphs, and in order to fingerprint them, they must be transformed into a byte stream. This process is called structure linearization. Algorithms for structure linearization depend on specific product data representations and show an inherent complexity. They are therefore further discussed in Anderl/Momberg (1998a).

The longevity of product data exceeds those of most other kinds of data. The long-term archiving of product data causes in fact problems on several levels. The

storage media on which the product data is deposited must be readable for the whole archiving time. The typical solution is to choose optical media, verify in defined intervals the readability of the data, and renew the media if they show data inconsistencies or if the playback and recording devices are no longer supported by the manufacturer. On the data representation layer the current practice is to store the data in proprietary formats like those of designated CAD systems. To overcome the problem of the data's incompatibility with newer versions of the application systems, either the system vendors are committed to guarantee backward data compatibility or the data is lifted manually to a newer version of the application system if the readability with the current version cannot be secured. A better solution is the use of standardized product data representations, of which the most prominent is ISO 10303 *Product data representation and exchange* (ISO (1994)), formerly known as *Standard for the exchange of product model data (STEP)*. Their apparent disadvantage that the diversity of state-of-art application features are not mappable into standardized representations (which is not necessary to satisfy the legal terms) is offset by the published documentation of the standard. A software for reading the archived data can thus always be written in the future. In this context the integration of digital signatures in product data is particularly meaningful when using standardized representations. One approach to integrate approval and contract signing in STEP is described in detail in Anderl/Momberg (1998b).

The long archiving time of product data makes particular demands on the security of the cryptographic algorithms used for signing. A typical parameter for the grade of an algorithm's security is the length of the keys, in the case of digital signatures both the length of the private/public keys and the length of the fingerprint. As the computing power available for cryptanalysis is not foreseeable for the lifetime of the product data, those lengths should be chosen carefully. "Be conservative. If your keys are longer than you imagine necessary, then fewer technological surprises can harm you." (Schneier (1996), p. 168) Schneier recommends key lengths above 128 Bit for lifetimes longer than a few decades. The state of art in cryptographic algorithms uses 1024 Bit for public key algorithms and 160 Bit for hash-functions. The decision concerning the key length must therefore be made individually.

Limitations on the key length are imposed by the export restrictions of various countries for cryptographic algorithms (at least if used for encryption). An internationally acting enterprise that maintains a globally distributed product development is therefore obliged to regulate the key management policy in its development projects.

3 PRODUCT COMPONENT AUTHENTICATION

The tagging of product components with part numbers, serial numbers, and type plates is as old as manufacturing itself. The objective is always to establish a

connection between the component and the product data describing it, be it a technical drawing, a derived product model like replacement part catalogs or service manuals, or the originating product data. Before discussing how cryptographic method may create a new quality of product component tagging, a classification of tagging methods shall be constructed.

A product component tag, in the following called tag for short, shows three layers of abstraction: the tag representation, the tag presentation, and the tag attachment.

Tag representation

The tag representation is formed by the tagging information itself. Each component in the product data owns an identification which describes either the whole series of components manufactured from its product data, the part number, or the individual instantiation of the components, the serial number. In numbering systems, both are typically constructed from a classification part, an identification part, and an informational part (Bernhardt/Bernhardt (1990)), and may be represented as a character sequence or purely numerical.

The classification part describes the component's integration in a classification system, from which the most prominent is the decimal classification, introduced by the librarian Melvil Dewey in 1876. Classification systems are usually defined in a company-wide manner. Its structure is therefore proprietary and shall not be discussed further.

The identification part identifies the component series, in case of a part number, or the individual component, in case of a serial number, uniquely. Frequently this will be an increasing number assigned by a central administrative department of a company.

The informational part allows humans to derive the component features directly. These include geometrical (length, diameter) and material (metal, alloy, plastic) information.

Tag presentation

The tag presentation determines how the tag information may be perceived. Beside human-readable presentations like a character string, a number or a color sequence, machine-readable presentations are in use, typically in combination with human-readable presentations to create a certain redundancy and to allow readability if the reading machinery is currently not available. Presentations that can be perceived by machines are ones suitable for optical character recognition (OCR) like the OCR-B font, bar codes like the European Article Number (EAN), or bit patterns like the DataGlyphs of Xerox Corporation. The presentation may also be influenced by the tag attachment method. For instance, when using magnetic ink or paint, a presentation like magnetic ink character recognition (MICR) may be chosen, or, when a transponder IC carries the tag information, the

presentation may be simply omitted because only the tag's representation needs to be embedded in the transponder's communication protocol.

Tag attachment methods

Almost every manufacturing process can be used to attach tagging information. Some are casting, etching, engraving, burning-in, embossing, glueing or coating. For manufacturing processes for which a form has to be provided, like casting, the attachment of serial-number-like information is typically difficult to perform. In this case part numbers or a different attachment method are used. Not only adhesive labels may be glued but also holograms which imply a particular presentation of the information possibly including 3D optical pattern recognition. Under coating the application of ink and paint, invisible (radioactive, ultraviolet fluorescent) or visible (magnetic, colored), is summarized too.

Most widespread are the embossing of alphanumerical information with letter stamps, the integration of tagging information in the casting form, the mounting of adhesive labels, and the riveting of type plates. More conservative but similarly widespread are the labelling of the components with pens, the attachment of labels with wires or cords, markings on the component packaging, or accompanying a paper describing the component.

A rather new method is the integration of integrated circuits (IC) in or their attachment on the components. The communication is accomplished either electrically via contacts or wireless by inductive or capacitive coupling. With this method more sophisticated information can be tagged because the presention is omitted, as mentioned above. Even information about the current condition of the component could be made available, if the IC were equipped with suitable sensors, which is summarized under the concept of intelligent components.

The selection of an attachment method is significantly determined by the requirements on the durability and resistance against removal attempts. For instance the application of cast-in tags may be preferred over adhesive labels.

Introducing authenticity in component tagging

Methods for strengthening component tagging against forgery may be introduced in all three of the above abstraction layers. They partly address the authenticity of the tagging information or additionally of the component itself.

On the representation layer the concept of the digital signature as described in the first part of this paper may be applied. The signature may replace the tagging information completely or supplement it to allow to differentiate between the component identification and the signature verification. The latter may be accomplished by applying the public key of the signer to the signature. The decryption not only firms the authenticity of the signature but also reveals the tagging information. The public key may be provided with service documentation which is increasingly distributed on digital media. This approach is particularly

interesting when using transponder ICs for tagging as they only deal with the representation of the tag.

The presentation of tagging information may be secured by hiding the information and must therefore be considered together with the attachment of the tagging information. A fairly new companion of cryptography, steganography, deals with the hiding of information in other information, in particular text, pictures, and audio data. The application on physical components is therefore not covered directly but some techniques can be derived. For example, the randomness of the least significant bits in raster pictures may be used to hide information. Transferred to physical components, the roughness of a non-effective part surface may code information, if the surface is intentionally and specifically distorted, for instance by laser or electro eroding. The coding can be set up in a more robust manner by generating a white noise, transforming it into the frequency space with the Fourier transform, adding the tagging information by deterministically changing coefficients, and transforming it back. The resulting pattern is applied to a non-functional surface in the same manner as above. The advantage of this approach is that partial destruction of the surface may still leave the information recoverable. The high expenditure makes such a method at least economical interesting for expensive and highly security relevant components.

Another approach could be to slightly but deterministically vary dimensions of part features that have no contribution to the parts functionality, thereby adding the tagging information. This must of course be accomplished outside of the tolerances of the manufacturing process.

The authenticity of an individual component can be achieved with adding a secret to it by simply keeping the method for authenticity verification secret. Such a secret could be the acoustical spectrum emitted by the part if excited by a certain frequency or frequency mixture. A second example is the explicit composition of the alloy the part is made of. It should be made clear that not the measurement result itself is the secret but the method by which it was gained, although the knowledge of the former could offer additional secrecy.

A second method is to identify characteristic parameters of the component series, measure these parameters for each individual component, and calculate a digital fingerprint of the data as described in the first part of this paper. The fingerprint may be signed and added to the tagging information. The set of characteristic parameters depends on the intended lifetime of component's fingerprint verifiability. For some components the verifiability must only be guaranteed until the component has been installed. In this case, parameters could be considered which were otherwise subject to signal-to-noise degradations due to attrition.

The set of approaches just presented shall be interpreted as an initial impulse for the development of further component and tagging information authentication methods. Then one additional secret may come from the unpredictability of the engineer's imagination.

For details on steganography see Wayner (1997). The yet single conference on information hiding was held in Cambridge, U. K. (Anderson (1996)).

4 CONCLUSIONS

Product data authentication constitutes a prominent ingredient to the establishment of a product's legally binding digital representation, the digital master, not only for enterprise-internal failure tracing but also for long-term archiving. The need for authentic product data can also be derived from the increase in the geographic distribution of product development processes combined with the use of combine-spanning intranets or the Internet and the resulting security issues. Thus the efforts for the integration of authentication methods in proprietary and standardized product data management and exchange structures should be intensified.

The authentication of product components is a fairly new discipline but will evolve swiftly, as the market for fake spare parts in some industrial sectors is spreading and the entailed loss of security in the product's operation will not be accepted by the customers.

The above remarks about product and product tagging authentication indicate that virtually any physical effect may be used to transfer and partially store tagging information. The time will tell how ingenuity of scientists and engineers combined with new manufacturing and measurement technologies may provide us with an increase of security in product usage and in the protection of intellectual and physical property.

5 REFERENCES

Anderl, Reiner ; Momberg, Martin (1997) *Authentisierung von Produktdaten*. Produktdatenjournal, v. 4, no. 2, pp. 41-45, 1997.

Anderl, Reiner ; Momberg, Martin (1998a) *Authentication of Product Data Structures*. Proceedings of the Product Data Technology Days 1998, Garston, Watford, 1998.

Anderl, Reiner ; Momberg, Martin (1998b) *Authentication of STEP Product Models*. Proceedings of the ProSTEP Science Days 1998, i. p., 1998.

Anderson, Ross (Ed.) (1996) *Information Hiding*. Proceedings of the First International Workshop, Berlin ; New York : Springer, 1996.

Bernhardt, Rolf ; Berhardt, Werner (1990) *Nummerungssysteme*. 2. Ed., Ehningen : Expert, 1990.

Diffie, Whitfield ; Hellman, Martin (1976) *New directions in Cyptography*. IEEE Transactions on Information Theory, v. IT-22, no. 6, pp. 644-654, 1976.

Dobbertin, Hans (1996) *Welche Hash-Funktionen sind für digitale Signaturen geeignet?* In: Horster, Patrick (Ed.): *Digitale Signaturen. Grundlagen, Realisierungen, Rechtliche Aspekte, Anwendungen.* DuD-Fachbeiträge, pp. 81-92, Braunschweig : Vieweg, 1996.

International Organization for Standardization (1994) *ISO 10303 Industrial automation systems and integration – Product data representation and exchange.* International Standard, ISO TC184/SC4, 1994.

National Institute of Standards and Technology (1993) *Secure Hash Standard.* NIST FIPS PUB 180, U.S. Department of Commerce, 1993.

RIPE Consortium (1992) *Research and Development in Advanced Communication Technologies in Europe (RACE): RIPE Integrity Primitives: Final Report of RACE Integrity Primitives Evaluation.* R1040, 1992.

Rivest, Ronald (1992) *The MD5 Message-Digest Algorithm.* Request for Comments 1321, Cambridge, 1992.

Schneier, Bruce (1996) *Applied Cryptography.* 2. Ed., New York : Wiley, 1996.

Wayner, Peter (1997) *Digital Copyright Protection.* Boston : Academic Press, 1997.

BIOGRAPHY

Prof. Dr.-Ing. Reiner Anderl studied mechanical engineering at the University of Karlsruhe, where he received his doctor degree in 1984 after working as a research assistant. In 1985 he took the position as a chief engineer and received habilitation in 1991 in Karlsruhe at the faculty of mechanical engineering. In April 1993 he received the professorship for computer integrated design at the TU Darmstadt. For several years he has participated in research on product development and standardization activities on international and national level.

Dipl.-Inform. Martin Momberg studied computer science at Darmstadt University of Technology. Since November 1993 he has worked as a research assistant at the department of Datenverarbeitung in der Konstruktion (computer integrated design). After establishing the IT infrastructure for the newly introduced undergraduate education in 3D-CAD design in mechanical engineering at TU Darmstadt, he now works on the integration of cryptographic methods such as digital signatures and steganography in the product development process, particularly in product data.