# Protecting PLM Data throughout their lifecycle

Rohit Ranchal, and Bharat Bhargava

Purdue University, Computer Sciences and CERIAS
West Lafayette, IN 47906, USA
{rranchal,bbshail}@purdue.edu

**Abstract.** Enterprises operate in a global economy with their operations dispersed across internal processes and external partners. Product Lifecycle Management (PLM) systems play a significant role in modern product development and management. There are multiple stages in product lifecycle that streamline by sharing data among PLM entities. Shared data may contain highly sensitive information such as trade secrets, intellectual property, private organizational or personal information. In large enterprise systems, it is difficult to understand and track data dissemination. Data sharing across global partners complicates and magnifies the problem further. The effect of shared data being leaked is one of the key risks. Existing approaches ensure security within the domain of an organization and don't address protection in a decentralized environment. We propose an approach for secure data dissemination using the Active Bundle scheme. This approach enables organizations to securely share information in their PLM steps and protects it throughout the product lifecycle.

**Keywords:** PLM, active bundle, data dissemination, security, privacy

## 1 Introduction

Modern product development is highly complex and increased competition has driven organizations to focus on core competences. A single organization can no longer efficiently handle product development and management in its entirety. Organizations focus on their expertise and outsource other activities to partner organizations specialized in these activities. This is done to achieve faster product development cycles, lower development costs and improve quality of service. Organizations rely on a complex web of distributed collaboration among internal business processes and external service providers to develop and deliver their products and services. This complex web of interactions is realized through Product Lifecycle Management (PLM) systems.

PLM is an information management solution that supports product development and management by streamlining the flow of product related information along all the stages of its lifecycle [1]. It provides a shared platform to connect various participating entities over the entire lifecycle of the product from concept to retirement. Product development is accompanied by many changes such as changes in customer demands, errors in design and planning, resource availability etc. Thus PLM deals with an enormous quantity of data flow. Effective collaboration and product management in a PLM system requires product data to be dynamically shared, readily

accessible and inherently secure. This includes all the product lifecycle stages that plan, build, manage, maintain, service and protect the product. Each stage involves interactions among entities that use available information, generate new information and share it further. Each entity defines specific information usage requirements before sharing it further. Being highly sensitive in nature, this information is not only necessary to develop and deliver products but is also responsible for improving efficiency, driving business decisions and maintaining competitiveness in the market.

### 1.1 Information Flow in PLM

Modern PLM systems engage global partners and employ cross-domain information exchange, where the information is dispersed across multiple entities and not under the control of a single owner. PLM aims to provide control over this information and tools to manage the overall lifecycle. Organizations like to keep track of their information flow to know how their information is used, with whom it is shared, and what actions are applied to it. It can be safely assumed that complete control over information flow, information usage and information tracking is possible in a trusted domain. Each entity in PLM has its own trusted domain but when data leaves this domain, there is very little or no visibility and control over this data. This data can be further shared with other entities in the PLM stages making it impossible for the entities in the earlier stages to track or know about its current state. The magnitude of data, complexity of processes and global distribution of entities in the PLM systems further complicate data sharing and dissemination. It is very difficult to support data sharing across partners and ensure its security at the same time. The threat of shared data being compromised is one of the key risks in PLM systems. Unauthorized information disclosures or data leakage can lead to huge financial and business losses and can be threatening to an organization's reputation. Thus, it is imperative that the information in PLM systems be securely shared and protected according to its owner's policies leading to trustworthy PLM systems.

### 1.2 Motivation

There have been numerous cases of product management systems being compromised resulting in substantial damages to the organizations. One such incident happened with Foxconn, which assembles about 40 percent of the consumer electronics products in the world. The hackers penetrated the Foxconn network and stole sensitive data including contact details of Foxconn’s global sales managers, usernames, IP addresses, client e-mails and purchases. This data could have been used to place fraudulent orders from the Foxconn’s clients. Foxconn had to take its services offline to prevent further damage [3]. Even the most reputed companies such as Apple, HP, Sony etc have shipped pre-owned laptops, hard drives, and other devices with viruses, worms, and trojans on them. A recent report published by Verizon indicates that there has been an increase in the number of hacking attacks and data breaches across the globe [4]. According to the report there were 855 data breaches in 2011 that involved more than 174 million compromised records. The report also

reflects the product development challenges for the organizations conducting global business. Organization insiders (the employees and contractors of an organization) account for 80% of computer and Internet crimes [5]. Security weaknesses in information dissemination are major threats that could be exploited by the malicious insiders [6]. PLM data resides in the globally connected networks and such incidents can have highly damaging consequences.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 describes the proposed approach. Section 4 presents the prototype implementation. Section 5 discusses the resilience of the proposed approach. Section 6 concludes the paper.

## 2 Related Work

Information in the PLM systems is scattered across various entities that participate in product development and management. One of the main challenges for PLM systems is getting complete control over the information. Tracking the information flow and ensuring its protection throughout its lifecycle is a significant issue with dynamic cross-domain information exchange. Each entity has specific security requirements. It is very difficult to know or compare the information security controls of actual and potential partners against the information owner's policies such as the capability of a partner to protect the information, level of protection actually being applied, outsourcing of information by the partner to other entities, compliance to regulatory and legal policies etc. Researchers have investigated issues with PLM in VIDOP project [7, 8]. In [9], the authors discuss security limitations of distributed PLM solutions. In [10], authors propose a solution based on the use of a Trusted Third Party (TTP) known as Management Entity. TTP is responsible for enforcing owner policies on the shared information. Substantial studies have been done to address information sharing and access control in the area of collaborative design in [11, 12] and supply chain in [2, 13, 14].

### 2.1 Issues with Current Solutions

Existing research lacks in addressing security issues with PLM. Available solutions typically operate in isolation and focus on data protection within the organization and do not extend to its partners. Security policies are defined to protect shared information, restrict access and regulate its usage. Each entity defines policies for its information. Examples of policies include: prevent data leaks to outsiders, enforce authorization to shared information, and control dissemination content. Information is protected according to the information holder's policies but there are often multiple information receivers for which the owner specifies multiple policies. These receivers can further disseminate the information. This requires policy communication, negotiation and enforcement in different (including unknown or untrusted) domains but the available solutions are unable to ensure the correct policy enforcement and control information dissemination in unknown or untrusted domains. Information

security standards are constantly evolving and there are no global security standards for PLM systems. Multiple, overlapping, disparate standards and the differences in the implementation of security controls add more complexity to the protection mechanisms. This can result in duplication of security controls and inefficiencies but more importantly may leave security gaps in the information flow [15].

Organizations do vendor comparisons, rely on service level agreements or contracts, perform audits, or hire specialized companies to identify vulnerabilities, detect violations and ensure conformance of business processes to specific security and privacy requirements. Efficient manual verification is possible on processes in the same domain and could only be performed on processes that are composed of small number of activities and generate less data. Such approaches don't work in unknown or untrusted domains and are insufficient to ensure data security throughout its lifecycle. We propose a data-centric approach that addresses the above-mentioned issues. It provides a security blanket around the information that traverses the PLM entities and protects it throughout its lifecycle. It enables PLM entities to receive the respective information without revealing extra information.

## 3 Proposed Data Sharing and Dissemination Mechanism

The main challenges in information sharing mechanisms are their limitations to provide control over data after it leaves the trusted domain. Common protection mechanisms consider data as passive entities that are unable to protect themselves. They require another active and trusted entity– a trusted processor, a trusted memory module, a trusted application or a TTP to provide protection and ensure correct policy enforcement in foreign domains. But trusting this entity requires taking risks. If the trusted entity is compromised, all the sensitive information is also compromised. Moreover, trusted entities (except TTP) should be installed in all the PLM entities, which may not be feasible for external partners. We propose a data-centric approach that transforms passive data into an active entity that is able to protect itself. It enables dynamic data dissemination decisions. The granularity of the data being shared with an entity is determined by the respective dissemination policy of the data owner. The proposed approach is based on the Active Bundle (AB) scheme [l6, 17].

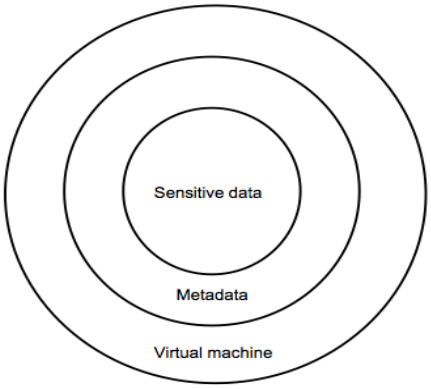


**Fig. 1.** Basic structure of an Active Bundle (AB).

### 3.1 Overview of the Active Bundle Scheme

The active bundle is a robust and an extensible scheme that can be used to securely disseminate data across multiple domains. An Active Bundle (more detailed description can be found in [16, 17]), the structure of which is shown in Fig. 1, is a data protection mechanism that encapsulates sensitive data with metadata and a virtual machine.

**Sensitive data:** It is the digital content that needs to be protected from privacy violations, data leaks and unauthorized disclosures. The digital content can include documents, pieces of code, images, audio, video files etc. The content in the sensitive data can have several sub-elements, each with different dissemination requirements. For instance, certain part of the data could be shared with marketing department such as product specifications, pricing etc and a different part of data could be shared with production department such as design documents.

**Metadata:** It describes the active bundle and its privacy policies. The metadata includes (but is not limited to) the following components (details available in [17]): (a) provenance metadata; (b) integrity check metadata; (c) access control metadata; (d) dissemination control metadata; (e) life duration value; (f) security metadata (including: security server id; encryption algorithm used by the Virtual Machine; encrypted pseudo-random number generator; trust server id used to validate the trust level and the role of a host; and trust level threshold required to access data in an active bundle); and (g) other application-dependent and context-dependent metadata. For instance, the access control metadata is used to ascertain the content for a specific receiver according to receiver's authorization.

**Virtual machine (VM):** It is the protection and policy enforcement mechanism that manages and controls the program code enclosed in a bundle. Its main functions include: (a) enforcing bundle access control policies through apoptosis (self-destruction) or data filtering (e.g., disclosing to a receiver only the portion of sensitive data that it is entitled to access); (b) enforcing bundle dissemination policies; and (c) validating bundle integrity.

### 3.2 Working of the Active Bundle Scheme

Unlike other solutions, the AB scheme does not require a client application on the receiver to execute its code. It can work like an applet, a jar file or a mobile agent. An active bundle is created with sensitive data, security policies and sent from one PLM entity to another. When arriving at the receiver entity, an active bundle ascertains the entity's trust level through a TTP. Using its disclosure policy, it decides whether the entity is eligible to access all or part of bundle’s data, and which portion of sensitive data can be revealed to it. An active bundle may realize that its security is about to be compromised. E.g., it may discover that its self-integrity check fails, or the trust level of the receiver entity is too low. In response, the bundle may choose to apoptosize, that is, perform atomically a clean self-destruction (one that is complete and leaves no traces usable for an attacker) [17]. An active bundle after completing its task on this entity travels to the next entity in the PLM chain. The information about next entity to

be traversed can either be specified in advance in the bundle (e.g. during its creation) or dynamically decided.

## 4 Active Bundle Prototype Implementation

The AB prototype has been implemented using the mobile agent framework Jade (http://jade.tilab.com/) [17, 18]. A mobile agent is a software object that contains code and carried data and is able to perform computations on visited hosts, transport itself from one host to another, and interact with and use capabilities of visited hosts [17]. The system contains: AB Coordinator; AB Creator; AB Destination; Directory Facilitator (DF); AB Services: Security Services Agent (SSA), Trust Evaluation Agent (TEA), and Audit Services Agents (ASA); and an AB. The components are distributed among Jade containers. Fig. 2 shows the GUI for agent management. It shows the containers and registered agents (DF, SSA, TEA, ASA) in the system. They can be setup on a single host or distributed on different hosts.

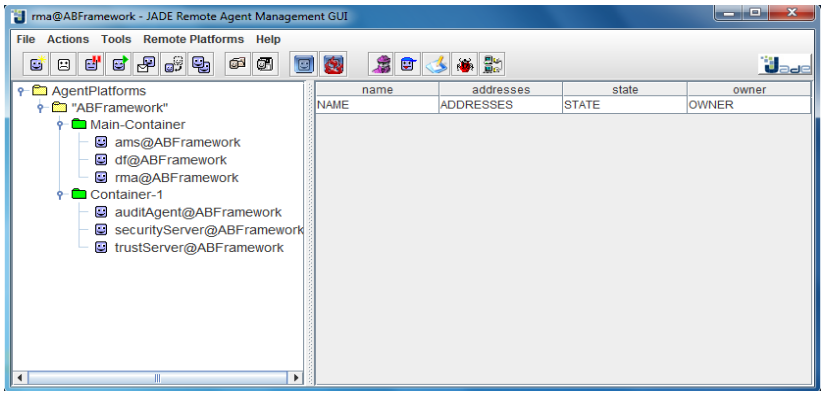


**Fig. 2.** AB Framework GUI using JADE.

**AB Coordinator** hosts the Jade DF, providing a yellow pages service. It is used by agents to register/deregister their services, and to search for services and destinations.
**AB Creator** is an application that accepts from a user, input including sensitive data, metadata, and the destination and transforms it according to the attributes of AB's structure and includes the code for VM. Next, it registers the AB with DF.
**AB Destination** hosts a container and receives ABs sent by the creator.
**AB Services** are TTPs implemented as three agents: (a) SSA that contains security related information about ABs. It is used for encrypting and decrypting ABs. Each AB is described in SSA using the following information: name, decryption key, and the threshold trust level that a receiver must satisfy to access data from AB. (b) TEA answers requests from SSA about the trust level of a specified host, which could be obtained using a trust management system [17, 20]. (c) ASA records and monitors activities of ABs. It receives audit information from ABs, and records this information for analysis by authorized entities (e.g., AB owners, or auditors) and to support dynamic metadata updates.

**AB** is a mobile agent constructed by AB Creator that has a set of attributes and operations.

### 4.1 System Functioning

With the assumption that there is a secure communication channel between the AB and the TTPs, below we provide a description of AB prototype functioning.

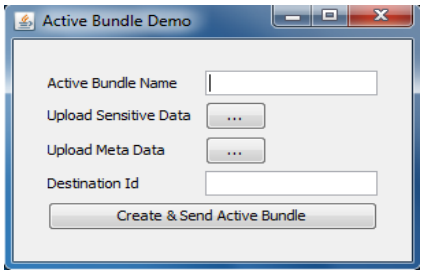


**Fig. 1.** AB creation GUI.

**Initialization of an AB:** An owner of sensitive data provides AB Creator with sensitive data and metadata as shown in Fig. 3. Sensitive data contains multiple versions of the content to support selective and controlled dissemination based on access policies and receiver authorization. Simple custom DTDs are defined to specify sensitive data and metadata policies in XML format. AB Creator constructs an AB by putting together data, metadata, and a VM. After this stage, the AB becomes an active entity (since it has its own VM) that can perform the remaining steps.

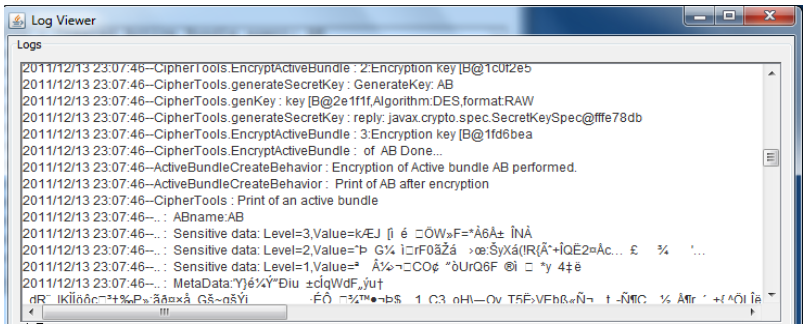


**Fig. 4.** AB processing logs GUI.

**Building an AB:** The steps taken in the process of building an AB are as follows (Fig. 4 shows the logs for this step):

- The AB gets two pairs of public/private keys from Security Service Agent (SSA) for each version of the sensitive data, where the first pair of keys is used for encrypting the data included in it and the second pair of keys is used for signing/verifying the data signature. The reason for having two key pairs is to prevent attackers from modifying AB's sensitive data and signing it again with

the private decryption key of the data owner.

- The AB sends a request to SSA asking it to record the AB's security information. The AB's identity data includes its name, decryption keys, and the trust level that a receiver must satisfy to use the AB. The goal is to keep the decryption keys and other auxiliary data for ABs in a trusted location. The decryption keys are given only to receivers that are eligible to access the AB.
- The AB computes a hash value for each version of the sensitive data and signs them using the signature keys. The signature certifies that sensitive data is from its owner.
- The AB encrypts sensitive data using the encryption key.

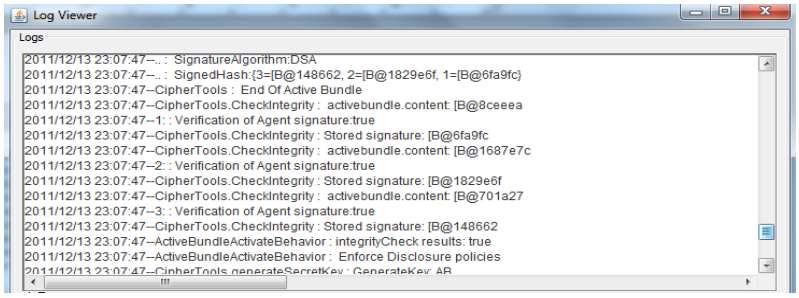


**Fig. 5.** AB processing logs GUI.

**Enabling an AB:** After arriving at the destination, AB enables itself (Fig. 5 shows the logs for this step). The steps of the enabling algorithm are as follows:

- AB sends a request to SSA asking for the security information on AB and the receiver's trust level.
- AB checks if the receiver's trust level is lower than the minimal trust level required for AB access. If so, the AB apoptosizes; otherwise, it executes the next step.
- AB checks integrity of its sensitive data. It computes the hash value for sensitive data and it verifies the AB's signed hash value by comparing it to the computed hash value. If verification fails, AB apoptosizes; otherwise, the AB decrypts the appropriate version of data according to access policies and receiver authorization.
- AB enforces its privacy policies and provides the data to the receiver.
- AB sends audit information to ASA. This information includes AB's name, the receiver's identity, and the name of the event being audited.

## 5 Resilience of the Proposed Approach

The current AB approach relies on TTP. This brings in all TTP related issues such as loss of control, lack of trust etc [18]. We are investigating approaches to decrease reliance on TTP making ABs more self-protected entities. One such approach based on predicates over encrypted data and multiparty computing for Identity Management is discussed in [18]. Another approach (used in Vanish [19]) is to use Shamir's

threshold secret sharing technique [21] to split decryption keys into N shares, and use a threshold t (< N) that defines the minimum number of shares required to reconstruct the key. The reconstructed key can then be used to decrypt the AB data. Key shares can be stored in a distributed hash table (DHT) system (such as Vuze), where each share is stored at a separate node. The main advantages of using DHT are decentralized and asynchronous communication and its large-scale geographic distribution making the practical attacks on key shares impossible. The malicious hosts can attack and alter the AB VM to gain unauthorized access to bundle's data. They can also deny the VM execution. In this case, data are not disclosed to unauthorized entities but legitimate entities are denied access to these data. The main challenge in implementing the AB's VM is assuring that a visited host executes the AB's VM code faithfully and correctly. We are investigating different approaches to address this. The idea is to intertwine the VM code and data together to make it incomprehensible and use obfuscation to hide data and program code within a scrambled code so that it still works, but provably cannot be reverse engineered [22].

## 6 Conclusion

PLM requires cross-company collaboration but sharing information externally raises numerous security concerns. Available solutions focus on data protection within the organization but do not address external information sharing. We need better technologies that provide security along with flexibility and adaptability to the PLM systems. The security mechanisms should not interfere with the existing information sharing and collaboration mechanisms. They should be able to protect shared information throughout its lifecycle. We propose the use of AB scheme that can be adopted in existing systems. AB security capabilities include protecting data throughout its lifecycle, self-protection on unknown/untrusted receivers, controlled data dissemination, selective data dissemination, activity monitoring, and dynamic policy adjustment. It enables organizations and their partners to share product information in a secure PLM environment with confidence that each participant's information is protected in accordance with its policies. The future work involves extending the prototype for AB scheme that doesn't rely on TTP, deploying it in a PLM system and performance evaluation of the system.

**Acknowledgement**
This research was supported by a grant from PLM Center of Excellence and CERIAS at Purdue University. Special thanks to Guneshi Wickramaarachchi and Lotfi Ben Othmane for their contributions to this research.

## References

1. Ameri, F., Dutta, D.: Product lifecycle management: closing the knowledge loops. Computer-Aided Design & Applications, 2(5), 577-590 (2005)

2. Atallah, M. J., Elmongui, H. G., Deshpande, V., Schwarz, L. B.: Secure supply-chain protocols. In: IEEE International Conference on E- Commerce, pp. 293-302, (2003)
3. Hackers attack Foxconn for the laughs, http://www.macworld.com/article/1165298/foxconn_reportedly_hacked_by_group_critical_of_working_conditions.html (2012)
4. Verizon: 2012 Data Breach Investigations Report, http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf?CMP=DMC-SMB_Z_ZZ_ZZ_Z_TV_N_Z037 (2012)
5. Carr, J.: Strategies and Issues: Thwarting Insider Attacks. Network Magazine (2002)
6. Cappelli, D., Moore, A., Trzeciak, R., Shimeall, T.J.: Common Sense Guide to Prevention and Detection of Insider Threats, V. 3.1. Carnegie Mellon University (2009)
7. Woerner, J., Woern, H.: Distributed and secure co-operative engineering in virtual plant production. In: Advanced Production Management Systems: Conference on Collaborative Systems for Production Management, 175-187, (2002)
8. Pels, H.: Federated Product Data Management in Multi-company Projects. Advances in Design, 281-291, (2006)
9. Leong, K. K., Yu, K. M., Lee, W. B.: A security model for distributed product data management system. Computers in Industry, 50(2), 179-193, (2003)
10. Rouibah, K., Ould-Ali, S.: Dynamic data sharing and security in a collaborative product definition management system. Robotics and Computer-Integrated Manufacturing, 23(2), 217-233, (2007)
11. Cera, C. D., Kim, T., Han, J., Regli, W. C.: Role-based viewing envelopes for information protection in collaborative modeling. Computer-Aided Design, 36(9), 873-886, (2004)
12. Brustoloni, J. C., Nnaji, B. O.: Intellectual property protection in collaborative design through lean information modeling and sharing. Journal of computing and information science in engineering, 6, 149, (2006)
13. Iyer, A. V., Ye, J.: Assessing the value of information sharing in a promotional retail environment. Manufacturing & Service Operations Management, 2(2), 128-143, (2000)
14. Cachon, G. P., Fisher, M.: Supply chain inventory management and the value of shared information. Management science, 46(8), 1032-1048, (2000)
15. Browne, N., Crespigny, M., Reavis, J., Roemer, K., Samani, R.: Business Assurance for the 21st Century: Navigating the Information Assurance landscape. Information Security Forum, (2011)
16. Ben Othmane, L., Lilien, L.: Protecting Privacy of Sensitive Data Dissemination Using Active Bundles. In World Congress on Privacy, Security, Trust and the Management of e-Business, pp. 202-213, (2009)
17. Ben Othmane, L.: Active Bundles for Protecting Confidentiality of Sensitive Data Throughout Their Lifecycle. Ph.D. Thesis, Western Michigan University, (2010)
18. Ranchal, R., Bhargava, B., Ben Othmane, L. Lilien, L., Kim, A., Kang M., Linderman, M.: Protection of identity information in cloud computing without trusted third party. In: 29th IEEE Symposium on Reliable Distributed Systems, (2010)
19. Geambasu, R., Kohno, T., Levy, A., Levy, H. M.: Vanish: Increasing data privacy with self-destructing data. In 18th USENIX Security Symposium, p. 56, (2009)
20. Bhargava, B., Angin, P., Ranchal, R., Sivakumar, R., Linderman, M., Sinclair, A.: A trust based approach for secure data dissemination in a mobile peer-to-peer network of AVs. International Journal of Next-generation Computing, 3(1), (2012)
21. Shamir, A.: How to Share a Secret. Communications of the ACM, vol. 22(11), pp. 612–613, (1979)
22. Heffner, K., Collberg, C.: The Obfuscation Executive. Information Security, vol. 3225, pp. 428-440, (2004)