ISSN : 0974 - 7435 Volume 8 Issue 2

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 8(2), 2013 [219-223]

Study of system security architecture

Jiang He

Department of Mathematics, Henan Institute of Science and Technology, Xinxiang. 453003, (R.CHINA)

ABSTRACT

This paper describes the hybrid nature of a DCE's communications and processing environment, including a discussion of the system security architecture that is present on each DCE member's node. This section illustrates the position of the TMS as a decision-making layer that supports the key management system (KMS) with assessments of trustworthiness.

KEYWORDS

Dynamic collaborative environment;
The key management system;
System security architecture;
Trust management system.

INTRODUCTION

Each member node contributed to the system security architecture, as shown in Figure 1. Each node executed a three-layered security agent that implemented this security construct. Some layers, like the KMS layer, contributed to the DCE at large, while others, like the Intrusion Detection System (IDS) layer, were focused more on the individual node. These agents were autonomous, in that the parameters were set by the operating node and not by network-wide security policies.

An agent-based approach was selected because of its suitability to a mobile collaborative environment[1].Each node possessed a complete security system and could operate independently based on peer nodes that were known to it or observations made first hand. A node could also join a coalition or collaborative group and take advantage of the group's information. The node retained this information when it chose to leave the coalition or the group's network area.

The KMS managed user identity certificates and established the rules for issuing, reissuing, and revoking certificates[2]. In a centralized network, this KMS relied on directory replication and certificate revocation lists (CRLs.) In a decentralized environment, the goal was to provide the KMS with access control decisions based on the trustworthiness of the perspective peer node.

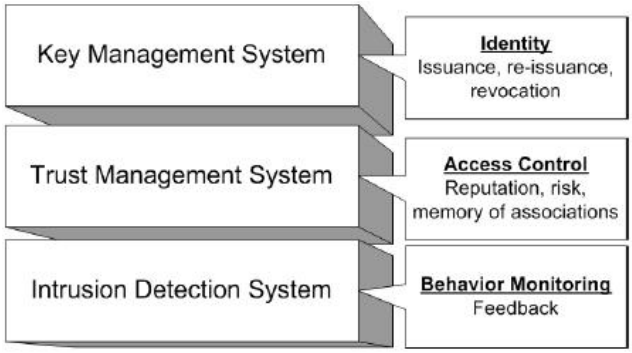

Figure 1 : System security architecture

The TMS was implemented as a central data-processing layer of the overall system security architecture. The TMS provided the KMS with a layer of abstraction of the overall trustworthiness of nodes, based on the activity of the nodes in the network. As the central layer, the TMS determined whether to trust or distrust its peers based on its individual trust thresholds.

The trust management system then reported its trust decisions to the KMS for its consideration.

At the lowest layer, an IDS or network monitoring scheme[3] provided periodic performance observations to the network. These observations were distributed throughout the system in a modified epidemic routing algorithm, similar to the selective dissemination scheme proposed by Datta[4]. The architecture's lowest level was simulated, as its specification and construction was beyond the scope of this paper.

The following sections develop the requirements for the trust management layer and detail the theoretical model underpinning its construction. First, we examine the requirements for building and using reputations in a virtual society or collaborative group. Then the TMS inputs and outputs are identified before the internal processes of the TMS are detailed.

IDENTIFYING TRUST MANAGEMENT SYSTEM REQUIREMENTS

Having established the location and general function of the TMS in the system security architecture, we looked at the inputs and outputs the TMS will require. In particular, we needed to identify the information necessary to collect, construct, and utilize reputations of peers within a virtual society. The eBay Feedback System (EFS) was examined as an example of a widely used reputation system to determine suitable system requirements[5]. The EFS was chosen because it enabled a behavior grading system in a large, well-documented environment. Through the eBay website, the EFS (eBay's Feedback System) aggregated positive and negative comments made by buyers and sellers to provide customers with some sense of the reliability or trustworthiness of a person they are considering doing business with.

The EFS introduced three features that were applicable to reputation management in general: positive and negative feedback, reputation aging, and identity. In the EFS, buyers and sellers left positive or negative feedback on each other's performance after conducting a transaction. Positive comments had the same weight as negative comments, meaning that a compliment had the same effect on a reputation as a complaint. A similar situation existed in a collaborative environment when two nodes participated in a file sharing or information exchange. Peers submitted positive feedback when a transaction was completed in line with their expectations. Transactions that were incomplete or unsatisfactory (e.g., the file was not as advertised, the service was too slow) resulted in the submission of negative feedback. The presence of both positive and negative feedback was deemed necessary for a complete reputation management system.

The use of feedback raised the requirement for the EFS's second feature. This feature was the need to age or fade feedback to prevent reputation gaming. Aging feedback diminished the impact older behavior feedback items (FIs) had on the reputation calculation. If the system did not age feedback, a comment made years ago had the same weight as a comment made on a current transaction. A malicious individual could take advantage of this weakness to build up a high reputation over time and then default or cheat without incurring much damage to his reputation. Aging FIs made this sort of attack more difficult because a user's performance had to be constantly maintained to sustain his positive reputation.

The third requirement observed in the EFS was identity. In the eBay system, a member's reputation rating reflected the number of other distinct members that have left feedback. Because the source of the FI was recorded, the EFS discarded duplicate items, hindering the opportunity for a single node to have undue influence over another node's reputation. For example, even if Alice left four positive (or four negative) comments on four distinct transactions, only one positive (or negative) item was added to Bob's reputation because all four pieces of input came from the same buyer. The EFS used login-password combinations to identify the user submitting feedback but other distributed systems used a PKI-based KMS to provide each member with a persistent identity[6].

In addition to the requirements derived from the EFS, a decentralized environment posed additional challenges for reputation management. Member nodes were not restricted to a single location or access point to obtain network services. Nodes could enter or leave network coverage and continue to operate in peer-to-peer mode. This characteristic was called nomadic membership.

When a node decided to establish an association with a new peer, there needed to be a procedure for each side of the transaction to establish the partner's identity and gain preliminary trust information without relying on a central authority or directory. This procedure was similar to the way individuals introduce themselves in social situations. Some systems[7] performed introductions by passing a reputation value or used a voting mechanism to extend trust to new associates. The node soliciting the introduction could not determine how or why the prospective associate had established a particular trustworthiness because of their lack of evidence to support the reported trust level. We concluded, therefore, that an independent determination of trust required a node to examine evidence of the prospective associate's behavior.

A more effective introduction process included a mechanism for the two prospective associates to share observed behavior history in such a way that they could derive the reputation of their prospective partner by having the proof to substantiate the given value. In our target environment, a node polled the Delegated Certificate Authorities (DCAs) and its Trusted Peers (TPs) for the new associate's identity certificate and behavior history. As a result, an effective reputation management system had to keep a certain number of its behavior observations so that it could provide non-reputable evidence to other nodes.

The preceding analysis examined both centralized systems (e.g., the EFS) and decentralized environments to collect requirements for a trust-based system. The following sections will discuss the sources for identities and behavior evidence, as they are external to the TMS layer. Internal mechanisms, such as reputation aging and the introduction process, will follow as part of the discussion on the TMS design.

ELEMENTS FROM THE KEY MANAGEMENT SYSTEM

In a well-connected and hierarchically organized DCE, the KMS had the ability to provide a control plane of authentication services. This ability was constrained by connectivity, lack of a naming policy, and the Dynamic Coalition Problem (DCP). Because of these constraints, there could be no expectation that all DCE members had verifiable identities, since not all members would be willing to surrender their autonomy to the DCE.

In many ways authentication presented the same requirements as authorization and was vulnerable to the previously mentioned constraints. Authentication required cryptographically verifiable credentials but the possession of identity credentials did not equate to verifiable permissions. The result was that the KMS was relied upon to handle identity credentials but that these credentials assumed less importance in a DCE than in a more controlled and organized environment.

The KMS, like the other layers of the system security architecture, resided on each node. Within each node, the KMS declared the node's identity to peers and established secure information exchanges with associates in the DCE. These associates were TPs and provided referrals to each other. Between associates, the KMS layer publicized the establishment and dissolution of associations to specially designated nodes called DCAs.

Within the security architecture, the KMS asked the TMS for trust assessments on specific users. The KMS provided the identity of a prospective associate and received a Go/No-Go assessment of that user's trustworthiness in return. These assessments allowed the KMS to accept or decline offers to associate with other users.

Identity imprinting

The concept of imprinting an identity on a network peer was borrowed from the world of biological sciences. The imprinting process required a node to declare its identity upon entering the network. This identity could have come from any one of three sources. First, the node's parent organization could have issued identity credentials before the node joined the DCE. Second, the node could have applied to another DCE member to issue credentials signed by a local DCA. Third, the node could have issued its own credential.

Because the KMS accepts the difficulty in verifying identity credentials, a node's declared identity was only used as an index for behavior grades. A user might create any number of aliases but these were all linked in some way to their declared identity. The user was discouraged from creating aliases by the reputation-scal-

FULL PAPER

ing mechanism.

Reports

The KMS recorded instantiations and dissolutions of trust within the system. Because of our target environment's use of IPSec to secure communications, these extensions of trust were implemented as security associations between peer nodes. Nodes notified the KMS of the association's status. The KMS then shared this information throughout the network. The notification messages were called reports to differentiate them from observations gathered from other nodes and were treated as trusted, global, information.

The KMS implemented two types of reports: registrations and complaints. Registrations and complaints were specially formatted messages that a node sent to notify the KMS of the establishment or dissolution of trust.

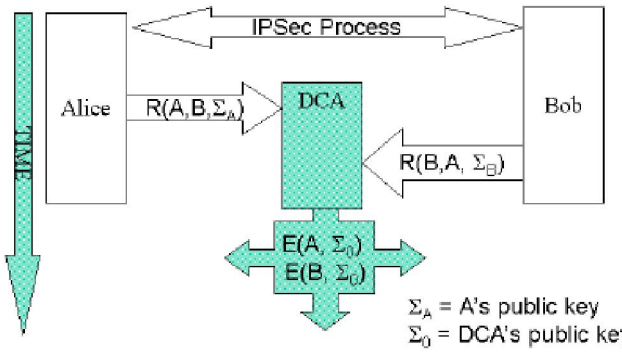

Figure 2 : Registering a security association

The registration message, illustrated in Figure 2, was specially formatted and signed by each node to provide non-repudiation. The message specified the identity of the nodes in the association and the signature of the node submitting the registration. Upon receipt, the DCA assigned the event message a serial number and broadcast two establishment messages to the network. The establishment message notified everyone in the network of the new association.

While the DCE was making its notification, both associates updated their list of TPs with each other's identification and began collecting behavior information on their new associate. Once the transaction was completed, both sides reported the dissolution of the association to the KMS. The dissolution message, illustrated in Figure 3, included an indication of each party's satisfaction with their partner's behavior during the transaction. This message type was treated as a signed event by the KMS for the purposes of auditing. Like the registration message, the dissolution message was broadcast to the network.

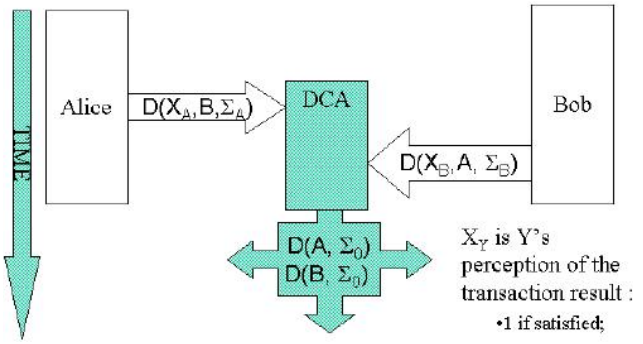

Figure 3 : Dissolution process

ELEMENTS FROM THE INTRUSION DETECTION SYSTEM

The IDS or network monitor provided periodic performance observations on peers in the network. The TMS informed the IDS of what to observe by providing lists of peer identities and contexts. Observations, it should be noted, were records of an individual node's expectations. Because observations stemmed from perceptions, they are not completely trusted but are used to confirm or augment reports received from the KMS.

The observations compared a node's expectations against the observed performance of its neighbors. Observations were made on trusted peers as well as on neighboring nodes that were within "listening range" but were not necessarily directly trusted. Nodes observed performance in areas such as resource sharing or file access and periodically generated positive or negative observations. A node received a "good" observation by doing what was asked of it. If Alice asked Bob for a file or to print an email and Bob agreed, Alice gave him a positive behavior observation that she shared with her other trusted peers. If Bob refused, she gave him a negative observation, regardless of the reason he had for refusing. If Bob did not answer her request, Alice could assume that he had moved out of range or was asleep, withholding any observation. This was acceptable because the TMS was an autonomous but not intelligent or rationalizing decision making system.

The observations contained the identifiers of the ob-

server and the observed, an observation, and the observer's signature, as shown in Figure 4. These observations were proliferated through the network in a modified epidemic routing algorithm, similar to the selective dissemination scheme proposed by, to spread information between trusted peers rather than flooding the network with observations.

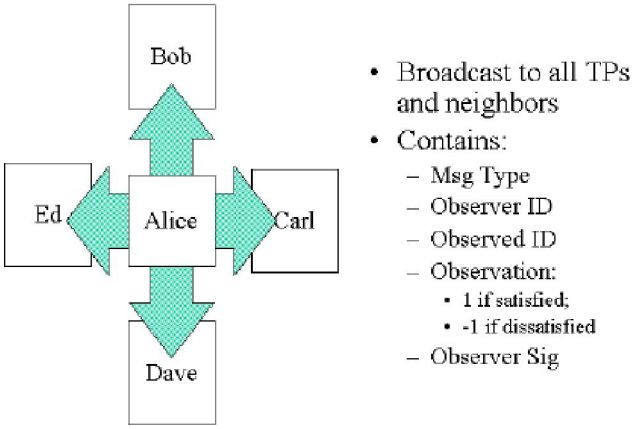

Figure 4 : Composition of a behavior observation

REFERENCES

[1] L.Bartram, M.Blackstock; Designing Portable Collaborative Networks. Queue, **1(3)**, 40-49 **(2003)**.

[2] G.C.Hadjichristofi, W.J.Adams et al.; A Framework for Key Management in a Mobile Ad-Hoc Network. International Journal of Information Technology, **11(2)**, 31-61 **(2005a)**.

[3] S.Buchegger, J.Y.Le Boudec; Nodes bearing drudges: towards routing security, fairness, and robustness in mobile ad hoc networks. Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, 2002., Canary Islands, 9-11 Jan 2002, 403-410 **(2002a)**.

[4] A.Datta, S.Quarteroni et al.; Autonomous Gossiping: A self-organizing epidemic algorithm for selective information dissemination in mobile ad-hoc networks. Technical Report, Ecole Polytechnique Federal de Lausanne, June **(2004)**.

[5] C.Keser; Experimental dames for the design of reputation management systems. IBM Systems Journal, **42(3)**, 498-506 **(2003)**.

[6] M.Thompson, A.Essiari et al.; Certificate-based authorization policy in a PKI environment. ACM Transactions on Information and System Security, **6(4)**, 566-588 **(2003)**.

[7] P.Dewan, P.Das}upta; Securing Reputation Data in Peer-to-Peer Networks. Proceedings of the International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), Cambridge, MA, 1-10 **(2004)**.