

Validation on the Enterprise SOA

Service Oriented Architectures, commonly known by the acronym “SOAs,” are changing the traditional security architecture of the enterprise. Transport layer security is being replaced by a more flexible message level security model, which does not carry the same level of assurance and trust. As web services’ based messaging becomes more ubiquitous, so does the need to provide efficient mechanisms for distributing and verifying message authentication. Today’s enterprise environments use a combination of transport and message level security to cobble together a security infrastructure. This is neither efficient nor scalable, and lacks the ability to provide long term audit ability of the transaction elements. We believe that SOA Validation is one of the single largest unmet needs in the enterprise environment for 2006.

Web services are driven by messaging which is inherently reusable. The messages are broken down into a group of elements, with each element capable of carrying signatures or encryption from disparate entities. A common message could take multiple hops and execute multiple transactions. Today’s enterprise environments, while passing identity assertions between parties, have no method to validate multiple signatures efficiently or historically. Enterprises are using traditional transport layer security models to assure the provenance of the last hop, but that does little to assure the third-party signatures on other elements of the transaction. This is a huge problem as SOAs become more prevalent and web services based transactions more common-place.

To address deploying an effective message level security model, we must address the defined membership of a web services implementation, or the aggregated membership of an enterprise SOA. We find, more often than not, that enterprises are supporting multiple divisional level web services and are not yet attempting to manage security at the corporate infrastructure level. We believe this will be the case for the next year as more web services are deployed, but long term SOA blueprints should plan to abstract security from the division level to the enterprise core. This will allow corporations to gain better control of audit and compliance, and most importantly “end to end” security cost.

Defining and managing a membership is a cumbersome task given traditional methods and technologies. We learned this in the late 90’s as we deployed public key infrastructure systems, tried to cross-certify them with trading partners, and then tried to manage them. Today, transport layer models simply do not provide the efficient throughput or the historic audit ability needed on the enterprise SOA. We solve the problem of massively distributed membership and membership validation differently,

and introduce a method for real time validation of signatures called the Trust Authority System (Trust Authority).

Trust Authority is a distributed system for data authentication using *authenticated dictionaries*. An *authenticated dictionary* is a data structure that supports authenticated membership queries; that is, in addition to returning the answer to a query, it also returns a cryptographic proof of that answer. A *distributed authenticated dictionary* architecture separates the data source (the host that maintains the membership) from the directory, which is local to each members web service. An overview of the basic message exchanges in a distributed authenticated dictionary is shown in Figure 1.

At the end of a specified time quantum, the source reliably pushes incremental updates of membership, as well as a digitally signed and time stamped fingerprint of the data structure, we call the *basis*, to all local directories. So when a member receives a web services message with an element or elements signed by other members, the validation query is local and returns the answer, a proof of that answer, and a copy of the signed basis from the source. Using these last two pieces of information, the user is then capable of verifying the authenticity of the response locally, while trusting only the basis that has been signed by the source.

This model allows for disparate directories to reside in trusted and un-trusted environments with little concern of compromise. This becomes important in a SOA environment where an enterprise can do little to affect the management and policies of a third party member's environment.

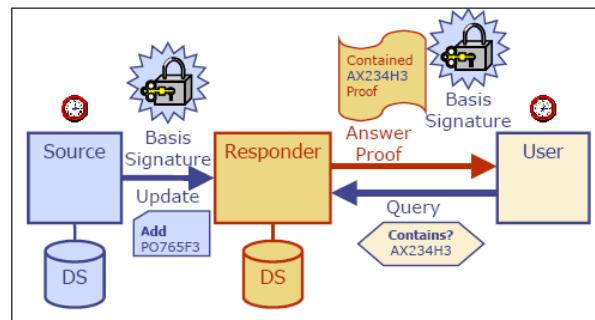


Figure 1 An overview of the protocol for authenticated distributed data systems. The source pushes updates containing a signed basis to the responder. The responder then answers user queries with a proof of the answer, and a copy of the signed basis from the source.

Our approach has many advantages, including efficiency (proving the validity of a query in 54 microseconds), centralized trust (the users have to trust only the source), distributed service (the directories are local to the web services querying the directory), low deployment cost (the directories do not require secure installations), and resiliency to denial-of service attacks (the validation authority does not answer queries itself). But, the most important element of Trust Authority is the historic

persistence and audit-ability we provide across the enterprise SOA. Applications of this technology exist in many domains, including content delivery networks, telecommunications, storage; and, trust on the enterprise SOA.

A schematic illustration of the Trust Authority architecture is shown in Figure 2. In Trust Authority, the *Basis* is a piece of data analogous to a fingerprint of the data structure. It is updated once per time quantum, digitally signed by the source (the to-be-signed basis, complete with timestamps and signing information, is called a *RichBasis*), and reliably pushed to all member directories. An *AuthenticResponse* is an object returned as the result of a query on an authenticated data structure; in addition to the answer, it contains a proof that can be verified against the basis to guarantee the integrity of the answer. The basis can be informally viewed as a hash of the entire data structure, which is archived with the validation authority for future forensic audit ability. A client verifies that the answer is authentic by hashing the query key and “combining” it with the proof; the result is equivalent to the signed hash of the entire data structure.

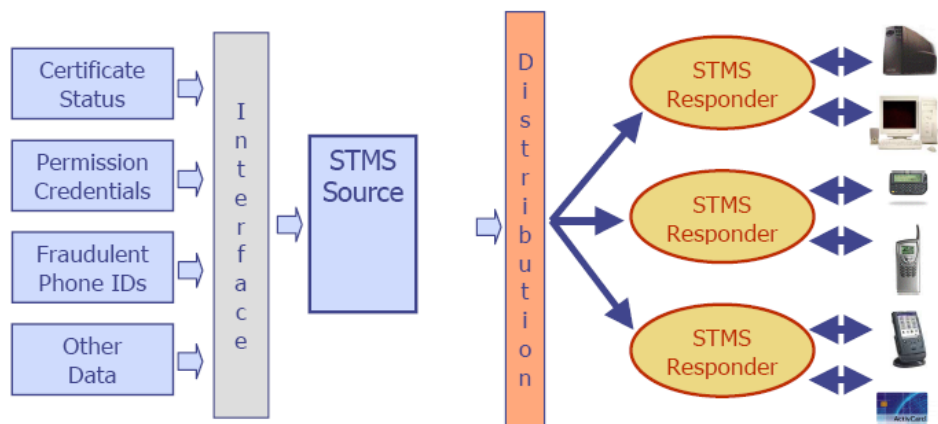


Figure 2 Overview of the STMS architecture, showing possible deployment models.

IAM Technology has integrated its Trust Authority model into a leading enterprise security appliance: *X10*. This central authority can be hosted by one or more members of a trading group to assure real time validation of signatures (en-mass) and historic validation of the specific elemental proofs. When we think of the enormity of the task of assuring compliance of authentication on a SOA in real time and with complete assurance, we see the need for a distributed approach to authentication and non-repudiation. We believe the Trust Authority model will become the standard for authentication within the enterprise SOA, not simply for efficiency of validation, but for its forensic capabilities assuring compliance and audit standards on the enterprise SOA.