The notion of trust management system was first introduced in the PolicyMaker system in 1996 by M. Blaze, J. Feigenbaum and J. Lacy. Trust management systems provide applications with a standard interface for them to check if the potentially dangerous operation conforms to their local security policy. They also provide users with a standard language for writing and specifying the policies and credentials. Trust management credentials describe a specific delegation of trust and include the role of public key certificates. A trust management system has five basic components: a method for describing actions, a mechanism for identifying principals, a language for specifying application policies, a language for specifying credentials, and a compliance checker.

KeyNote is a simple, flexible, and state-of-the-art trust management system. KeyNote provides a single and unified language for both local policies and credentials. KeyNote policies and credentials are called assertions. Assertions contain predicates that describe the trusted actions permitted by the holders of specific public keys. When a dangerous action is requested of a KeyNote-based application, the application submits a description of the action along with a copy of its local security policy to the KeyNote interpreter. KeyNote then approves or rejects the action according to the rules given in the application’s local policy.

The efficient policy management scheme, KeyNote has been applied to IPsec, a suite of protocols for network-layer confidentiality and authentication. The IPsec protocol itself does not define how to manage the policies for how protected traffic should be handled at security endpoints. Thus, to avoid applications being forced to duplicate at the application or transport layer cryptographic functions that already provided at the network layer, a policy management scheme for IPsec that based on the principles of trust management is needed.

There are two issues that problems of the security policy decisions in IPsec can be categorized into. First, it is required to make security decisions on packet-by-packet basis, hence necessitating the use of packet-based policy filtering. Second, IPsec policy control need two hosts to discover and negotiate the kind of traffic they are willing to exchange. The authors used a two level policy specification hierarchy to control IPsec traffic. At the packet level, the proposed scheme uses an efficient but less expressive filtering language that provides the basic control of traffic. It then controls packet filters through a more expressive but less efficient trust-management language.

The authors evaluated their scheme by modifying the OpenBSD’s IKE and showed the performance of the system efficient. This part is nice so we know that the system could really be useful for specifying and controlling authorization. However, since the scheme makes decisions as whether to approve or reject, it is actually a decision management system rather than a trust management system.