10-2004

Trust Establishment in Large Scale Grid Settings

Bo ZHU
National University of Singapore

Tieyan LI
Institute for Infocomm Research

Huafei ZHU
Institute for Infocomm Research

Mohan S . KANKANHALLI
National University of Singapore

Feng BAO
Singapore Management University, fbao@smu.edu.sg

See next page for additional authors

DOI: https://doi.org/10.1007/978-3-540-30208-7_46

Trust establishment in large scale grid settings

Bo Zhu[1,2], TieYan Li[2], HuaFei Zhu[2], Mohan S. Kankanhalli[1], Feng Bao[2], and Robert H. Deng[2]

[1] National University of Singapore, {zhubo, mohan}@comp.nus.edu.sg
[2] Institute for Infocomm Research, {zhubo, litieyan, huafei, baofeng, deng}@i2r.a-star.edu.sg

Abstract. Trust establishment is hard in grid architecture by the *ad hoc* nature. To set up trust in large scale of network is more difficult. In this paper, we propose an automatic key management (AKM) model and corresponding key construction schemes. The hierarchical structure is formed automatically and scale seamlessly in arbitrary network sized. Regions are configured differently according to various levels of risks faced. The novel model provides an integrated solution for self-organized trust establishment, upon which rich appliances are securely supported. It is automatic, flexible, and scalable. Furthermore, simulation results show that computation costs due to the variations are very small under common threshold and region size settings.

1 Introduction

Computational grids are a collection of heterogeneous computers and resources spread across multiple administrative domains with the intent of providing users easy access to these resources. Typical characteristics of grids include simultaneous use of large numbers of resources with dynamic requirements, use of resources from multiple administrative domains, complex communication structures, and stringent performance requirements. Under easier intentional attacks, security in this settings becomes more crucial and is eagerly needed.

To protect the grid infrastructure, Globus Toolkit [1] has been developed (current version GT3.0) and Grid Security Infrastructure (GSI) [4] is the *de facto* security standard in grid community. However, as indicated in [9], GSI suffers from many potential security drawbacks such as uncontrolled delegation, leaky infrastructure and insecure services. Thus, more security mechanisms must be developed to complement GSI and finally ensure inter-grid trust establishment. In the literature, several well known trust models have already been proposed [5, 3, 12]. All of these trust models are designed for certain scenarios, however, none of them is suitable to be used in large scale of grid environment efficiently.

In this paper, we proposed an automatic key management (AKM) model, which is hierarchical and formed dynamically. We carefully study the joining and leaving behaviors to design a scalable scheme. In addition, to achieve flexibility and adaptability, a new concept-*Regional Trust Coefficient* (RTC) is introduced to indicate the security condition of a region. To demonstrate our scheme, we do simulations and analyze the results.

The rest of the paper is organized as follows. In Section 2, we introduce the notations, definitions and general design of AKM. In Section 3, we describe in detail how AKM handles adjustments to secret shares subject to common node-based and region-based operations. Simulation results are given in Section 4. Section 5 studies the related work. At last, we conclude the paper.

2 Automatic key management architecture

2.1 A self-organizing architecture

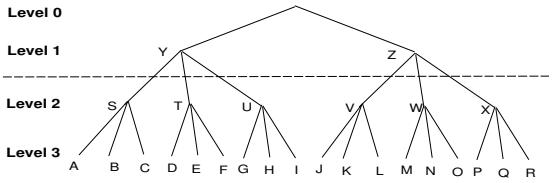


Fig. 1. A Three-level Hierarchical Model

Fig. 1 depicts a hierarchical model of our architecture. The main idea of AKM is that we maintain a relatively static upper level controlled by an offline root *Trusted Authority* (TA) to meet the requirement of virtual group, at the same time, we maintain dynamic regions flexible enough to handle nodes' joining/leaving. Although the regions are highly dynamic, we can set appropriate RTCs so that the occurrences of region-based operations is low. If we assume that the first $L+1$ levels (from level 0 to level L) of AKM are static, and the lower levels are dynamic. For a node denoted as X on level L, we call all the sub-regions originated from it as the *Dynamic Domain* (D-Domain) of X.

2.2 Notations and definitions

Informally, we introduce the definitions of some concepts used in this paper.

- *Real Nodes*: the leaf nodes in the hierarchical structure of AKM. They are corresponding to real devices, and have their own PKI key pairs.
- *Virtual Nodes*: the branch nodes in the hierarchical structure of AKM. They are virtual and thus do not represent real devices in the region.
- *Master Node*: the branch node that a node originates directly from.
- *Region*: consists of all the real and virtual nodes originating directly from the same virtual node. Each virtual node corresponds to a region.
- *Overall Region Size (ORS)*: the number of nodes which possess secret shares originated from the same secret, i.e. the secret key of the same region, between two consecutive secret share updates of this region.

– *Regional Trust Coefficient (RTC)*: the security parameter indicating the security level of a region. It is defined as the ratio of the threshold of a region to its ORS.
– *Global Trust Coefficient (GTC)*: the global lower limit on RTC.

In this paper, we use the following notations (shown in Table 1):

Table 1. Notations in AKM

h	the height of the global key tree	ID_i	the identifier of a node "i"
PK_i	the public key of a region "i"	SK_i	the secret share held by node "i". If node "i" is virtual, the secret share is also its secret key
pk_i	the public key of a real node "i"	sk_i	the secret key of a real node "i"
$SK_i(M)$	a message M is signed by SK_i	q	a large prime number
p_n	be the probability of a node being compromised	p_r	be the probability of a region being compromised

2.3 Secret sharing process

In AKM, all the region's secret keys originate from one global secret key either directly or indirectly. The secret sharing process based on Shamir's threshold scheme [8] is performed recursively from root down to the lowest level. At each round, a region's secret key either is destroyed after secret shares originated from it have been distributed to all the members of the region, or has not ever been explicitly generated. In AKM, the threshold of a region is invariable during its lifetime, because the calculation of changing the configuration of a threshold scheme from (n, k) to (n, k') is much higher than that of changing from (n, k) to (n', k). The secret share held by a region is used as its secret key, and its public key is generated from the secret key. Unlike virtual nodes (i.e. regions), each real node generates its own PKI key pair through some other method.

During the initialization of AKM, an offline root TA controls the key distribution process from level 0 to level L, and publishes public keys of all the static regions. However, it has no knowledge of what happens from level $L + 1$ downwards. Namely, D-Domains are transparent to the root TA. In each D-Domain, public keys of all the sub-regions are published to all the nodes within the domain. Therefore, a real node needs to store two kinds of public keys: (1) public keys of static-level (level 0 to level L) virtual nodes which are published by the offline root TA; (2) public keys of dynamic-level (level $L + 1$ onwards) virtual nodes within the D-Domain to which the node belongs.

It should be absolutely clear that the proposed AKM scheme does not requires a preset hierarchical structure for all the nodes. Instead, AKM only requires a small number of nodes during the initialization. More importantly, the structure of the whole network may vary when more nodes join or leave, and such variation

does not require any TA. In addition, each region can set or change its own size and threshold of secret sharing, as long as its RTC is not less than GTC.

3 Node-based and Region-based Operations

Common node-based and region-based operations include "Join", "Leave", "Merge", "Partition", "Expansion", and "Contraction". The first two are executed in the same way as [6], and here we concentrate on region-based operations only.

3.1 "Merge" and "Partition"

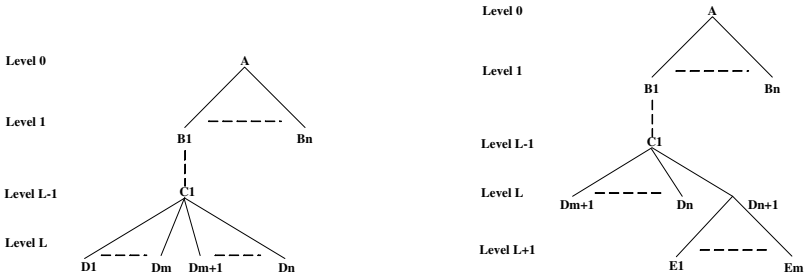
"Merge" operation happens when the number of nodes within a region drops under its threshold. Then, the region is divided into a few parts and each part is combined into one nearby region. Since the thresholds of the target regions are invariable, "Merge" operation can be viewed as a series of "Join" operations.

"Partition" operation happens when RTC of a region drops under GTC or is lower than the security level expected. A straight-forward solution is to partition the region into two regions with almost the same size. For example, region B_i with size $2n$ and the threshold k is partitioned into two regions B_i and B_{m+1}, each of which has n nodes and keeps its threshold as k. For the n nodes remaining in region B_i, they just need to renew their secret shares after the "Partition" operation. In order to assign new secret shares to the n nodes in region B_{m+1}, firstly, region B_{m+1} chooses k regions at level 1, and then selects k nodes from each of the k regions. Without lost of generality, we denote the group of the k regions , the group of the k nodes from region B_j $(j = 1, \cdots, k)$, and the group of all these k^2 nodes as $G_B = \{B_1, \cdots, B_k\}$, $G_j = \{C_{j1}, \cdots, C_{jk}\}$, and $G = \{C_{11}, \cdots, C_{1k}, \cdots, C_{k1}, \cdots, C_{kk}\}$, respectively. Then, a "Partition" request signed by the secret key of region B_i is generated and multicasted to all the nodes in group G. The IDs of region B_i, B_{m+1}, and the k regions are sent together with the request.

By Lagrange interpolation, we have $SK_{B_i} = \sum_{j=1}^{k} SK_{B_j} l_{B_j}(ID_{B_i}) \ (mod\ q)$ and $SK_{B_j} = \sum_{h=1}^{k} SK_{C_{jh}} l_{C_{jh}}(0) \ (mod\ q)$, where $l_{B_j}(ID_{B_i}) = \prod_{r=1, r\neq j}^{k} \frac{ID_{B_i} - ID_{B_r}}{ID_{B_j} - ID_{B_r}} \ (mod\ q)$ and $l_{C_{jh}}(0) = \prod_{r=1, r\neq h}^{k} \frac{ID_{C_{jr}}}{ID_{C_{jr}} - ID_{C_{jh}}} \ (mod\ q)$. Combining them, we have $SK_{B_i} = \sum_{j=1}^{k} \sum_{h=1}^{k} SK_{C_{jh}} l_{C_{jh}}(0) l_{B_j}(ID_{B_i}) \ (mod\ q)$. Similarly, we get $SK_{B_{m+1}}$. Therefore, we know that $SK_{B_{m+1}} - SK_{B_i} = \sum_{j=1}^{k} \sum_{h=1}^{k} SK_{C_{jh}} l_{C_{jh}}(0) R_j \ (mod\ q)$, where $R_j = l_{B_j}(ID_{B_{m+1}}) - l_{B_j}(ID_{B_i})$.

To help a node with identity $C_{(m+1)i}$ in region B_{m+1} generate the new shares of $SK_{B_{m+1}}$, each node C_{jh} in group G computes the partial share $SK'_{C_{jh}} = SK_{C_{jh}} l_{C_{jh}}(0) R_j + \sum_{r=1}^{k-1} a_{jhr} ID^r_{C_{(m+1)i}} \ \ (mod\ q)$, where $a_{jhr} \in Z_q, for\ r = 1, \ldots, k-1$. Then distributes the partial share to node $C_{(m+1)i}$. After receiving k^2 partial secret shares from the nodes in group G, node $C_{(m+1)i}$ adds them together to get a share of $SK_{B_{m+1}} - SK_{B_i}$, denoted as $SK'_{C_{(m+1)i}}$. According to the homomorphic property, each node in region B_{m+1} can compute its new share of $SK_{B_{m+1}}$ by adding $SK'_{C_{(m+1)i}}$ to its original share of SK_{B_i}.

3.2 "Expansion" and "Contraction"



(a) Before "Expansion" (b) After "Expansion"

Fig. 2. "Expansion" Operation

"Expansion" Operation happens when RTC of a region drops under GTC or when AKM reaches its capacity upper limit (namely, RTCs of all the regions in AKM are equal to GTC). For example, as shown in Figure 2(a), before "Expansion" the level of AKM is L. We assume that, original secret sharing of region C_1 is executed by $SK_{D_i} = a_0 + a_1 \cdot ID_{D_i} + \cdots + a_{k-1} \cdot ID_{D_i}^{k-1} \pmod q$.

Now there is a new node that wants to join, but the RTC of region C_1 is already equal to GTC. Therefore, "Expansion" operation is executed.

Firstly, chooses a group of m nodes from region C_1 which will be degraded to Level $L+1$, where $k \leq m \leq n-k+1$. Without loss of generality, let the group be $G = \{ID_{D_1}, \cdots, ID_{D_m}\}$. Then, selects k nodes from group G. Without loss of generality, let the group be $R = \{ID_{D_1}, \cdots, ID_{D_k}\}$. Following that, chooses a new identity denoted as $ID_{D_{n+1}}$ for the master node of all the nodes degraded. According to Shamir's threshold scheme [8], the secret share for node D_{n+1} can be calculated by the k nodes in group R. However, during "Expansion" operation, this secret share would be never calculated out or recovered explicitly, since we do not assume the existence of a TA at this stage.

For simplicity, the same $(n,\ k)$ threshold scheme is employed in the newly created region D_{n+1}. Without loss of generality, we assume that a new identity ID_{E_i} is assigned to the node whose old identity is ID_{D_i}, or the identity is chosen by the node itself. Then each node in group R calculates the partial secret share denoted as SK'_{E_i} by $SK'_{E_i} = SK_{D_j} \cdot l_{D_{n+1}} + \sum_{r=1}^{k-1} b_{jr} \cdot ID_{E_i}^r \pmod q$, where $l_{D_{n+1}} = \prod_{h=1,h\neq j}^{k} \frac{ID_{D_{n+1}} - ID_{D_h}}{ID_{D_j} - ID_{D_h}}$, and distributes it to the node whose new identity is ID_{E_i}. Finally, each node recovers its new secret share in the new region by combining all the secret shares: $SK_{E_i} = b_0 + b_1 \cdot ID_{E_i} + \cdots + b_{k-1} \cdot ID_{E_i}^{k-1} \pmod q$, where $b_0 = SK_{D_{n+1}}$, $b_r = \sum_{j=1}^{k} b_{jr}$ $(r = 1, 2, \ldots, k-1)$. Therefore, all the coefficients of the secret sharing polynomial of the new region are cooperatively determined by the k nodes.

"Contraction" Operation happens when AKM reaches its capacity lower limit. To ensure that the number of nodes in any region in AKM is not less than the threshold set before, we have to decrease the level of the structure. Similar to "Merge" operation, "Contraction" Operation can be viewed as a series of "Join" operations as well.

4 Simulation Analysis

Intuitively observed, as in highly dynamic environment, lots of regions with small size result in rapid transform of the structure of the key tree. On the other hand, if their sizes are too big, we may have problems with intra-region routing. Current on-demand routing protocols, such as AODV [7] and OLSR [2], handle well when the size is around 100 to 250 nodes. Thus, it is suitable to set the region size within this range.

4.1 Regions with the Same RTC

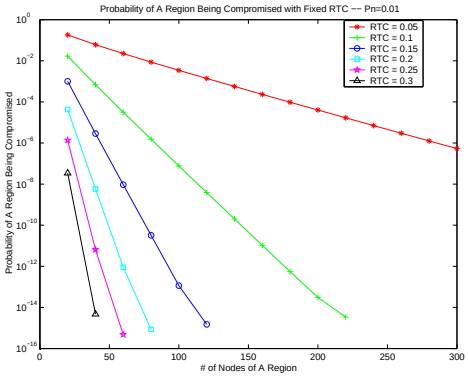


Fig. 3. P_r with fixed RTC

As shown in Figure 3, say a single curve, we can see that the higher the RTC, the smaller the value of p_r. Therefore, it is suitable to use RTC as an approximate index of the security condition of a region. In addition, observing the bunch of curve, the higher the RTC, the faster the value of p_r decreases. Due to the two properties, we say that a higher RTC is good for the sake of security. However, it leads to a higher threshold which consumes more computation power.

At the same time, we find that, when both RTC and the region size are small, this region is easy to be compromised. For example, in Figure 3, given that the p_n is 0.01 and the RTC of a region is 0.05, p_r may be higher than 0.01 when the size of the region is less than 75. Fortunately, we need to be aware of this only during the initialization of a region. Thereafter, if nodes leave the

region, compared to the decreasing of the region size, the increase of the RTC can bring more positive effects on the security condition of the region. For instance, suppose a region has 40 nodes at the beginning, and its RTC is 0.05. According to Figure 3, p_r is 6.07%. After 20 nodes' leaving the region, the region size drops to 20. However, because its RTC increases to 0.1 at the same time, we find that p_r decreases to 1.69%. It means that this region is more secure than before.

4.2 Computational Costs of Region-based Operations

Since both "Merge" and "Contraction" operations can be viewed as a series of "Join" operations, our simulation focuses on "Partition" and "Expansion" operations. We run the simulation on a Pentium III 800 laptop.

Table 2 shows the computation cost of "Partition" and "Expansion" operations under different ORSs and thresholds. In the tables, GPSS stands for time for generating partial secret shares for all the nodes in the newly generated region, while GNSS stands for time for generating the new secret share. Simulation results show that the computation cost of both "Partition" and "Expansion" operations is quite small under common threshold and region size settings. For example, when the ORS of a region is 100 and its threshold is 15, it only takes 31 miliseconds to complete the "Partition" operation. As to the "Expansion" operation, the total cost is less than 8 miliseconds.

Table 2. Computation Costs of "Partition" and "Expansion" Operations

		"Partition" Operation (msec)			"Expansion" Operation (msec)		
ORS	Threshold	GPSS	GNSS	Total	GPSS	GNSS	Total
100	5	8.87	0.03	8.90	1.59	0.01	1.60
100	10	18.59	0.11	18.70	4.52	0.02	4.54
100	15	30.96	0.28	31.24	7.91	0.02	7.93
100	20	46.24	0.60	46.84	11.69	0.02	11.71
100	25	62.00	0.80	62.80	16.42	0.03	16.45
250	10	47.63	0.11	47.74	11.00	0.03	11.03
250	20	124.20	0.68	124.88	27.67	0.03	27.70
250	30	205.27	1.29	206.56	54.71	0.07	54.78
250	40	302.27	2.24	304.51	90.16	0.07	90.23
250	50	422.73	3.77	426.50	135.78	0.07	135.85

5 Related Works

X.509 [5], SPKI [3], and PGP [12] are three of the most well-known trust models. The X.509 trust model [5] is centralized that each participant has a certificate signed by a central CA. Since GSI employs X.509 certificates, this trust model can be used within a grid domain. SPKI trust model [3] is more flexible by supporting

delegation certificate. But how to control the proxy/delegation certificates is still an unsolved problem. PGP [12] is a distributed trust model that builds one's trust from its neighbors.

Technically similar with our approach, in [11], Zhou and Haas focused on establishing a secure key management service in an ad hoc networking environment. They proposed to use threshold cryptography to distribute trust among a set of servers. Their solutions are only suitable for a small group of servers and are inefficient given large scale networks. In [6], Kong et al. extended the scheme in [11] to normal nodes. It minimizes the effort and complexity for mobile clients to locate and contact the service providers. One major weakness of this scheme is that it becomes either inefficient or insecure with increasing participants. In [10], a more extreme case, where there is no CA at all, was considered. Each user is its own authority domain and issues public-key certificates to other users.

6 Conclusion

In this paper, we proposed a hierarchical trust establish model, which is efficient, flexible and scalable. By building a dynamic key management scheme, we achieve strong security without much tradeoff on efficiency. Furthermore, simulation results show that our scheme is very efficient under common threshold and region size settings.

References

1. Globus toolkits v3.0 of the globus project. Http://www.globus.org.
2. Thomas Clausen Et. Al. Optimized link state routing protocol, September 2001. Internet draft, draft-ietf-manet-olsr-06.txt.
3. C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory, September 1999. RFC 2693.
4. I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proceeding of 5th ACM Conference on Computer and Communications Security Conference*, 1998.
5. R. Housley, W. Polk, W. Ford, and D. Solo. Internet x.509 public key infrastructure certificate and certificate revocation list (CRL) profile, 2002. RFC 3280.
6. Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, and Lixia Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *IEEE 9th International Conference on Network Protocols (ICNP'01)*, 2001.
7. Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA'99*, 1999.
8. A. Shamir. How to share a secret. *Communications of the ACM*, Vol. 22(11):612–613, November 1979.
9. Mike Surridge. A rough guide to grid security, 2002.
10. Srdjan Čapkuny, Levente Buttyán, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. Technical Report 2002/34, EPFL/IC, May 2002.
11. Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network Magazine, Special Issue on Network Security*, Vol. 13, No.6, 1999.
12. Phil Zimmermann. Pretty good privacy (PGP), October 1994.