Trust Credential Distribution in Autonomic Networks

Tao Jiang
Institute for Systems Research
University of Maryland
College Park, MD 20742
Email: tjiang@umd.edu

John S. Baras
Institute for Systems Research
Department of Electrical and Computer Engineering
University of Maryland
College Park, MD 20742
Email: baras@umd.edu

***Abstract*—Autonomic networks are networks that are self-organized with decentralized control and management. Accurate trust establishment and maintenance is essential for secure and reliable message transmissions in autonomic networks. Because of the mobility and dynamics, trust management in autonomic networks is a much more dynamic problem than in traditional server-based networks. In traditional networks, centralized trusted servers provide necessary trust credentials for trust establishment. However, such servers are not available in autonomic networks. Trust credentials are distributed among users in the network. In this paper, we develop and analyze distributed schemes for efficiently and securely distributing trust credentials, so that users are able to establish reliable trust relations with their neighbors. Our approach is inspired from network coding, where trust documents are combined during distribution.**

I. Introduction

Autonomic networks are networks that are self-organized, and controlled and managed in a decentralized way. Accurate trust establishment and maintenance is essential in such networks. Trust relations between neighbors are necessary to make sure that the transmitting messages are not leaked to the enemy. On the other hand autonomic networks pose several formidable challenges for the establishment, control and management of trust relationships due to lack of infrastructure and centralized servers. Because of the mobility and dynamics of autonomic networks, new trust relations need to be established when users meet new neighbors, and old trust relations need to be updated as well. In addition, the validity and value of trust credentials change with time as well as with environmental conditions and scenario stage. Thus trust establishment and maintenance in autonomic networks is a much more dynamic problem than in traditional server-based networks.

In traditional networks, trust management relies on centralized control servers, such as trusted third parties (TTPs) and authentication servers (ASs). Those servers are trusted and available all the time. Prior works within this framework ([1], [2], [3]) all assume an underlying hierarchical structure within which trust relationships between ASs are constrained. However, such servers are not available in autonomic networks, and trust credentials are distributed among users in the network.

To establish trust in such a distributed way has several advantages. It saves network resources (power, bandwidth, computation, etc.), which are limited in wireless mobile environments. It avoids the single point of failure problem as well. Moreover, the networks we are interested in are dynamic with frequent topology and membership changes, and distributed trust has the desired emergent property [4], as users only contact a few and easy-to-reach users. However, it also poses difficulties and new challenges for trust management systems.

In autonomic networks, trust credentials are typically issued and often stored in a distributed manner. Most of existing works in trust management (e.g. [5]) assume that one has already gathered all the potentially relevant credentials in one place and does not consider how to gather and disseminate these credentials. The assumption that all credentials are stored in one place is inconsistent with the idea of decentralized trust management. Credential distribution raises many interesting questions. When Alice wants to assess trustworthiness of Bob, where should she look for credentials? Often, she cannot look everywhere. How can she efficiently obtain requested credentials? In addition, what are the best places to store the credentials, so that they can be easily located, well protected and timely updated?

Trust credential distribution provides the foundation for trust evaluation. After the user obtains necessary trust credentials for the target user, he applies an evaluation policy to draw conclusions about the trustworthiness of the target user. Trust evaluation policy design has been an active field of research. Different policies or rules have been developed and studied, such as [6], [7] and [8]. Although choosing the right model to evaluate trust and obtaining credentials to compute trust go hand in hand, trust credential distribution is fairly independent of the specific evaluation model. In this paper, we propose a trust credential distribution scheme that uses linear network coding [9] to combine credentials during transmissions. The proposed scheme is inherently different from the commonly used request-response approach, and is absolutely decentralized, which nicely handles the dynamic nature of the problem.

This paper is organized as follows. Section II reviews related work on trust credential distribution and P2P file sharing. A general description of trust credentials is provided in Sec. III. Section IV describes our network coding inspired trust credential distribution scheme in details. The performance of our scheme is discussed in Sec. V with simulation results.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

Section VI concludes the paper and discusses future work.

II. Related Work

As one of the first works in the literature, Eschenauer in [10] proposed a trust establishment scheme based on Freenet [11], where trust credentials are routed and searched using distributed hash tables (DHT). More specifically, for a particular trust credential, its ID is mapped into a value using a universally defined hash function. This hash value also corresponds to a unique node ID in the network. Then the credential is routed to and stored in this node. When searching for this particular credential, the requester calculates the hash value using the same hash function and sets its request destination as the corresponding node. In this Freenet-based scheme, trust credentials are routed by hash-based routing instead of flooding.

The problem of trust credential distribution shares many characteristics of distributed peer-to-peer (P2P) file sharing systems, such as [12] and [11]. Trust credentials are files to be shared for trust management. Thus many trust management systems consist of components that are inspired from P2P file-sharing systems. For instance, the trust management scheme P-Grid [13] is based on scalable replication of tree structures, which is the framework for some P2P systems. As mentioned above, Eschenauer [10] uses hash-based routing in one of popular P2P networks Freenet for distribution of trust credentials. Request routing in Freenet avoids flooding and improves with time. Files, or trust credentials in this context, are replicated by caching at every node, which causes information to converge to where it is most needed.

Despite the similarities between trust credential distribution and P2P file sharing, there exist several unique properties of trust credential distribution. The size of trust credentials are a lot smaller compared to the hundreds of million or billion bit files in P2P sharing. The requests in trust credential distribution usually aim at multiple credentials, for instance, a request may ask for all trust credentials about the trustworthiness of node A. While in P2P one request explicitly points to one single file. Furthermore, trust credentials change much more frequently than files because of various reasons. For instance, trust levels may decrease with time, new evidence may be discovered, or behaviors of the target may have changed. In this paper, we propose a new trust credential distribution scheme that uses *network coding*, and we show that the new scheme handles the above unique properties and performs well in terms of efficiency and security. Network coding has been used in P2P file sharing as well. Gkantsidis and Rodriguez designed a P2P file sharing system named Avalanche, in which intermediate peers produce linear combinations of file blocks as in network coding in [14]. Acedanski et al., independently of Gkantsidis and Rodriguez, analyzed random linear coding based storage in [15].

III. Trust Credentials

Different trust contexts require different types of credentials. Examples might be your driver's license, your social security card, or your birth certificate. Each of these has some information identifying you and some authorization stating that someone else has confirmed your identity. Some credentials, such as your passport, are important enough confirmation of your identity that you would not want to lose them, lest someone uses them to impersonate you.

In cyberspace, users rely on digital trust documents. A digital trust document is data that functions much like a physical credential. In this paper, we focus on digital trust documents. Here are some examples of digital trust documents:

- *Digital certificate*: which is issued by a certificate authority or entity and verifies that a public key is owned by a particular entity. Digital certificates are used to thwart attempts to substitute one person's key for another. A digital certificate consists of three things: a public key, certificate information ("identity" information about the user, such as name, user ID, and so on) and one or more digital signatures. Digital certificates are widely used in PGP and X.509.
- *IAFIS* (Integrated Automated Fingerprint Identification System): which uses human fingerprints as identities. It has made law enforcement officials capable to easily share information about criminals and quickly compare a suspect's fingerprint image with millions of similar imprints.

The trust credentials can be generated by some trusted party as in centralized networks (e.g., Certificate Authority), by friends who know each other, or by certain monitoring schemes ([16], [17]). Creation of trust credentials is not in the scope of this paper.

In autonomic networks, information is usually partial and incomplete. Uncertainty of trust must be introduced in the trust credentials, which is usually represented by a value. This value denotes the degree of trust the issuer of the credentials has on the target user. The value can be binary-valued, either trust or distrust, or multiple-valued, such as four levels of trust in PGP [18], or even continuous in an interval, say $[-1, 1]$.

In dynamic environments, the validity and value of trust credentials change over time and space. Every user has confidence values on the credentials he stores. The confidence value depends on several factors, for instance, time elapsed since the credential is issued, or communication distance taken by the credential to reach the user. When the confidence value is below certain threshold, the corresponding credential is considered to be invalid.

IV. Network Coding Based Scheme

As we mentioned in Sec. I, our scheme uses linear network coding to combine credentials during transmissions. Network coding is a recent field in information theory proposed to improve the throughput utilization of a given network topology. Ahlswede et. al. [19] were the first to introduce network coding. An overview of network coding and a discussion of possible Internet applications are given in [20]. In this section, we describe our proposed scheme for trust credential distribu-

tion that produces efficient propagation of trust credentials and the associated operations are purely decentralized. .

We assume a population of users (*trustors*) that are interested in the trustworthiness of a particular user (*trustee*). The trust credentials about the trustee are initially stored in a number of users scattered throughout the network. These users are the issuers of the trust credentials, or they have retrieved these credentials from others. In the rest of the paper, we only consider trust credentials about one single trustee. All the operations are applied on those credentials about the particular trustee. Each trust credential, in the form of a digital document, has a unique ID, which is obtained by applying a universally defined hash function to the document. Our scheme can be easily extended for multiple trustees.

In autonomic networks, users do not directly communicate with all other users; they only communicate with a small subset of users, which we call the neighborhood. In our scheme, users frequently check with their neighbors for new credentials. The new credentials are forwarded to the node, once it discovers them in its neighborhood. Whenever a user forwards trust credentials, it produces a linear combination of all the credentials it currently stores and the combined documents it has received from its neighbors.

We represent the network as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Edges are denoted by $e = (v, v') \in E$, in which $v = \text{head}(e)$ and $v' = \text{tail}(e)$. The set of incoming edges is denoted as $I(v) = \{e \in E : \text{head}(e) = v\}$, and the in-degree of v is $deg_I(v) = |I(v)|$. Similarly, the set of outgoing edges is denoted as $O(v) = \{e \in E : \text{tail}(e) = v\}$, and the out-degree of v is $deg_O(v) = |O(v)|$.

We denote by $X_l^v, l = 1, \ldots, h$ one of the h credentials the node v stores and by $W_{e'}^v$ the combined document transmitted to v via edge e'. Consider one outgoing edge of v, edge e. The combined document transmitted via link e is defined as

$$W_e = \sum_{l=1}^{h} a_{l,e} X_l^v + \sum_{e':\text{head}(e')=v} b_{e',e} W_{e'}^v. \qquad (1)$$

All these operations take place in the finite field $\mathbf{F}_q$. The coefficient $a_{l,e}$ and $b_{e',e}$ are randomly picked from $\mathbf{F}_q$ and the data content of each trust document is represented by d elements from $\mathbf{F}_q$.

We define the coefficient vector of a combined document W_e as $C_e = [c_{e,1}, c_{e,2}, \ldots, c_{e,m}]$, where m is the total number of trust credentials available. According to Eqn. (1), the coefficient of credential X_l^v is $a_{l,e}$. Suppose the coefficient of a credential $X_{l'}$ in the combined document $W_{e'}^v$ is $c_{e',l'}$. Then we have that

$$c_{e,l'} = \sum_{e':\text{head}(e')=\text{tail}(e)} b_{e',e} c_{e',l'}.$$

Figure 1 depicts the flow of documents from edge e' to node v to edge e.

Observe that given m distinct trust documents, a user can recover them after receiving m combined documents for which the associated coefficient vectors are *linearly independent* from each other. The reconstruction process is similar to solving linear equations.

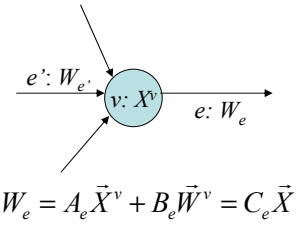

Fig. 1. Flow of documents

Our network-coding based scheme involves only local interactions. Users need not be aware of the existence of any trust credential and its location. They only contact their neighbors to check whether there are new credentials or new combined documents. This is a great advantage as compared to any request-response scheme, such as the Freenet-based scheme. Request-response schemes require routing tables. If the topology of the network changes, new nodes come in or some nodes move out, routing tables need to be updated immediately. While, our scheme adapts quickly to the topology changes, since users only contact their current neighbors. In addition, request-response schemes require that users know the document IDs before sending out requests, which requires global information exchange. Our scheme operates without knowing any document ID.

V. Performance

In this section, we focus on the performance of our network-coding based trust credential distribution scheme.

A. Effectiveness

We first address a key question: how efficient the credential distribution scheme is? We consider the case where a trustor, which is represented as node v in the network, requests m trust credentials for a particular trustee, denoted as $X_1, X_2, \ldots, X_m$. The in-degree of v is $k = deg_I(v)$. We assume the network operates in discrete time[1]. At each time instant, nodes transmit one combined document to each of their outgoing neighbors. Thus v receives k random linear combinations of the m documents $W_1, W_2, \ldots, W_k$ with the corresponding coefficient vectors associated with each of the k combined documents. We define C_j as the coefficient vector for the combined document j at node v, where $C_j = [c_{j1}, c_{j2}, \ldots, c_{jm}]$.

Let $rank_k$ denote the rank of the matrix composed of the k coefficient vectors, which is also the dimension of the subspace spanned by the vectors. $rank_k = d$ means that there are d independent vectors among the k vectors. Since all the coefficients in the k vectors are randomly picked

[1]Systems operating in discrete time require synchronization, which is difficult to achieve in autonomic networks. We assume synchronization for ease of notation. The results can be easily extended to the asynchronous case by denoting each transmission event as a time instant.

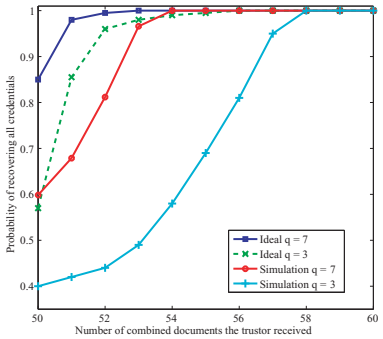

Fig. 2. Probability of recovering m credentials vs. number of documents received

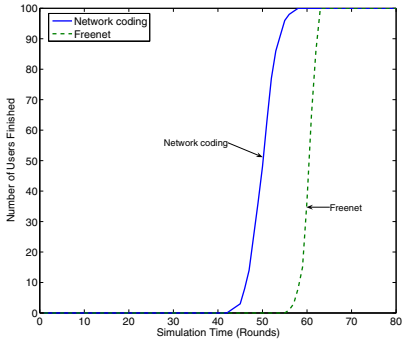

Fig. 3. Number of nodes finishing over time

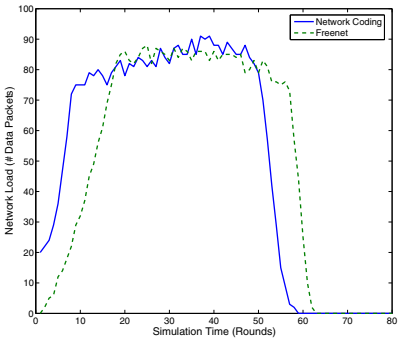

Fig. 4. Network load over time

from the finite field $\mathbf{F}_q$, if the first l columns of the matrix $[C_1, C_2, \ldots, C_k]$ are linearly independent, they span a vector space of dimension l and size q^l. The probability that the next column avoids this space is $1-q^l/q^m$. Therefore, we have that the probability of $rank_k = r$ has the following form [21]:

$$\Pr[rank_k = r] = \frac{\prod_{j=m-r+1}^{m}(1-q^{-j})\prod_{j=k-r+1}^{k}(1-q^{-j})}{q^{(m-r)(k-r)}\prod_{j=1}^{r}(1-q^{-j})}. \quad (2)$$

Therefore, the dimension of the coefficient vectors the trustor gathers depends on k, m and q.

We also study the effectiveness of our scheme by simulations. To study the performance of relatively large number of users, we have implemented our network coding based scheme in MATLAB. There are 64 nodes randomly placed in the 1×1 square. If the distance between two nodes is less than or equal to 0.25, they considered to be neighbors. 50 trust credentials are randomly placed on one of the nodes in the network. The simulations were run in rounds. At each round, every node checks updates of their neighbors. If there is update on one neighbor, a new combined document is created by the neighbor and sent to the node. A node is randomly selected as the trustor who tries to recover all the 50 credentials based on all the combined documents it has received. We ran the simulations with two finite fields of different orders ($q = 3$ or 7). The results are shown in Fig. 2, where the calculated results from Eqn. (2) are also plotted for comparison. With the scheme using the finite field of higher order it is easier to recover credentials because the probability of selecting independent vectors is higher. The simulation results are lower than the calculated results because the paths that the combined documents pass through are dependent. Thus, the coefficients are correlated. Nevertheless, the trustor can recover all 50 documents with less than 60 combined documents (just 10 more than the necessary number), which shows that our network-coding based scheme is efficient.

We also compare the performance of our trust credential distribution scheme with the Freenet-based approach in [10]. We define the *finish time* as the number of rounds that a trustor can recover all the trust credentials she requests. In this simulation, 20 distinct credentials are randomly stored in these nodes. All nodes are trustors who want to obtain these 20 credentials.

In Figure 3, we plot the finish time of each node, where the x-axis represents the simulation time and the y-axis represents the number of nodes that have finished. The performance of the network coding based scheme is better compared to the Freenet based scheme. Because by combining the credentials with network coding, within a few rounds, credentials can be spread throughout the network. While for the Freenet based scheme, just one credential is transmitted per round, and the search phase at the beginning takes long time.

Figure 4 gives the load of the network with time. The total network load of the two schemes is close to each other. While, the network coding based scheme converges much faster than the Freenet based scheme.

B. Heterogeneity

We have mentioned that there are different types of trust credentials with various confidence values. In addition, the importance of these credentials varies dramatically. In other words, the trust credentials in the network are *heterogeneous*. A good trust credential distribution scheme should take such heterogeneity into consideration. Credentials with high confidence values and of high importance should have high priority in transmission and they should be recovered and updated faster than ordinary credentials.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

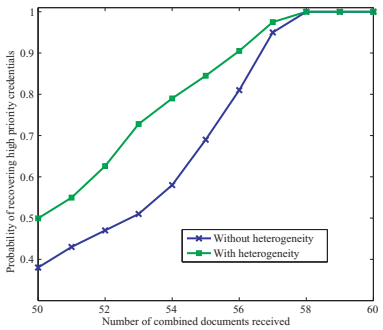

Fig. 5. Probability of recovering all high priority credentials vs. number of documents received

In our scheme, we propose to separately combine these high priority credentials. All high priority credentials are marked. If node v discovers a high priority credential or a high priority credential is updated at time t, it immediately sends out new combined documents to all its outgoing edges in the following way:

1) Check the latest document node v transmitted to edge e, denoted as $W_e(t')$ at time t'.
2) Create a new combined document $W_e(t) = aX_H + bW_e(t')$, where X_H is the high priority credential. Node v transmits $W_e(t)$ to edge e immediately.

This way, only two linear independent coefficient vectors of $W_e(t')$ and X_H are needed to recover X_H, instead of waiting for the entire m independent coefficient vectors. Thus high priority credentials have much higher chance to be recovered and updated. If more than one high priority credentials are discovered at node v, one combined document is created for each high priority credential. We ran the simulations by setting 20 out of 50 credentials as high priority credentials. Figure 5 shows the comparison of recovering high priority credentials with and without differentiating credentials. The modified scheme in this section helps to recover and update high priority credentials much faster.

VI. Conclusions

In this paper, we presented a scheme to distribute trust documents in autonomic networks based on network coding. The proposed scheme is purely decentralized with no global information exchange and very easy to implement in a self-organized manner. We proved that the scheme is efficient and adapts fast to network changes.

As future work, we plan to analyze the performance under different network topologies and settings and further explore the influence of mobility and malicious nodes. In this work, we assume that transmissions are perfect. However, in mobile ad hoc networks, transmission failures and errors are very common. It is in our future plans to investigate how resilient our scheme is to failures and errors.

Acknowledgment

This work is prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. Research is also supported by the U.S. Army Research Office under grant No DAAD19-01-1-0494.

References

[1] J. Steiner, C. Neuman, and J. I. Schiller, "Kerberos: An authentication service for open network systems," in *USENIX Workshop Proceedings, UNIX Security Workshop*, 1988, pp. 191–200.
[2] V. D. Gligor, S.-W. Luan, and J. Pato, "On inter-realm authentication in large distributed systems," in *Proceedings of the IEEE Conference on Security and Privacy*, 1992, pp. 2–17.
[3] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: Theory and practice," *ACM Transactions on Computer Systems (TOCS)*, vol. 10, no. 4, p. 265 C 310, November 1992.
[4] V. D. Gligor, "Security of emergent properties in ad-hoc networks," in *Proceeding of the Security Protocols Workshop*, Sidney Sussex College, Cambridge, UK, April 2004, pp. 256–266.
[5] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The role of trust management in distributed systems security," *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, pp. 185–210, 1999.
[6] M. K. Reiter and S. G. Stubblebine, "Authentication metric analysis and design," *ACM Transactions on Information and System Security*, vol. 2, no. 2, pp. 138–158, 1999.
[7] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*. ACM Press, 2004, pp. 1–10.
[8] T. Jiang and J. S. Baras, "Trust evaluation in anarchy: A case study on autonomous networks," in *Proceedings of Infocom*, Barcelona, Spain, April 2006.
[9] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, February 2003.
[10] L. Eschenauer, "On trust establishment in mobile ad-hoc networks," Master's thesis, Electrical and Computer Engineering Department, University of Maryland, College Park, 2002.
[11] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, p. 46, 2001.
[12] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001, pp. 149–160.
[13] K. Aberer, "P-Grid: A self-organized access structure for p2p information systems," in *Proceedings of 9th International Conference on Cooperative Information Systems*, 2001, pp. 179–194.
[14] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proceedings of IEEE INFOCOM*, vol. 4, Miami, FL, March 2005, pp. 2235–2245.
[15] S. Acedanski, S. Deb, M. Mdard, and R. Koetter, "How good is random linear coding based distributed networked storage?" in *NetCod*, 2005.
[16] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. Boston, Massachusetts, United States: ACM Press, 2000, pp. 255–265.
[17] Y. Zhang, W. Lee, and Y.-a. Huang, "Intrusion detection techniques for mobile wireless networks," *ACM/Kluwer Wireless Networks Journal (ACM WINET)*, vol. 9, no. 5, pp. 545–556, September 2003.
[18] P. Zimmermann, *PGP User's Guide*. MIT press, 1994.
[19] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
[20] P. Chou, Y. Wu, and K. Jain, "Network coding for the internet," in *Proceedings of IEEE Communication Theory Workshop*, Capri, 2004.
[21] J. M. V. Lint and R. M. Wilson, *A Course in Combinatorics*. Cambridge, 1992.