A Family of Collusion Resistant Symmetric Key Protocols for Authentication

Bruhadeshwar Bezawada and Kishore Kothapalli

Center for Security, Theory, and Algorithmic Research
International Institute of Information Technology
Gachibowli, Hyderabad 500 032, India
Tel.: +91-40-23001967 Exts:143,151
bezawada@iiit.ac.in, kkishore@iiit.ac.in

Abstract. We address the problem of message authentication in communication networks which are resource constrained or are performance bound. Recent research has focused on development of symmetric key protocols for authentication in such networks. In these protocols, the sender generates a pool of keys -used to sign the messages, and distributes a different subset of keys -used to verify the signatures, to each user. However, in these protocols, users can collude to combine their keys and impersonate the sender by generating the sender signatures. In this work, we describe a family of collusion resistant symmetric key distribution protocols for authentication which address the problem of collusion. We show that the collusion resistance achieved using our protocols is practical (and hence, sufficient) for networks whose communication diameter is known or is within fixed bounds. Furthermore, we show that some existing protocols in literature are members of our family of protocols.

Keywords: Authentication, Collusion resistance, Key Distribution.

1 Introduction

In communication networks, many critical tasks require regular exchange of information among the nodes in the network. In such scenarios, motivated by economic gains or other malicious intent, the data may be manipulated by a single node or by a group of such intermediate nodes. The propagation of such incorrect data can lead to instability of networks, loss of service, loss of revenue and damage the reputation of service providers. In particular, data manipulation by a group of colluding nodes is a serious problem as the extent of damage possible is higher than that is possible by attacks launched by a single malicious node. Thus, protecting the integrity of the data against attacks launched by a group of colluding users is an important problem in communication networks.

A standard solution to achieve collusion resistant message authentication is to employ digital signatures using public-key cryptography. A sender signs the message with its private key and the receiver can check the authenticity of the signature using the public key of the sender. However, the cost of signature

S. Rao et al. (Eds.): ICDCN 2008, LNCS 4904, pp. 387–392, 2008.

generation and verification in public-key crypto systems is relatively high for network devices and can slow down per-packet processing considerably.

To address the computational overhead in public-key crypto systems, several works [1,2,3] have proposed using symmetric key protocols for achieving message authentication. The sender uses a shared symmetric key to sign the message and the receiver verifies the signature using the same key. As most protocols [1,4] rely on multiply shared keys for reducing sender and user storage, collusion is likely in such protocols. However, in earlier works, the issue of collusion resistance has not been studied in detail.

In this paper, we address the issue of collusion resistance in symmetric key distribution protocols such as those in [1]. Towards this, we propose a family of collusion resistant symmetric key distribution protocols for message authentication in a communication network. Our contributions are as follows:

- We show that our key distribution protocols provide sufficient collusion resistance against message tampering for networks whose communication diameter is known to be within certain bounds, say, $O(\log N)$. Higher collusion resistance can be achieved by increasing the sender storage marginally. Although, our protocols require higher storage at the sender, the storage at the receivers is still $O(\log N)$ as in [1]. Furthermore, we show that, the currently known solution [1] is a member of our family of protocols.
- We show that, for most practical networks, the sender can choose to store a smaller number of keys and hence, reduce the signature cost per packet.

Organization. In Section 2, we describe the problem in detail, outline our network model and assumptions. In Section 3, we describe our family of collusion resistant symmetric key distribution protocols and provide a detailed analysis of their collusion resistance. In Section 4, we present the experimental results of applying our protocols to networks with different diameters. Finally, in Section **??**, we conclude the paper and outline some future work.

2 Problem Description

We address the problem of message authentication in communication networks. As an example, consider the link state routing protocol on the Internet (e.g., OSPF) that requires a router to broadcast link updates in its neighborhood to the entire network. This information is critical as other routers recompute their routing tables using this information. Since the information passes through different routers, one or more malicious routers can manipulate the information for selfish gains. Hence, there is a need for message integrity preserving mechanisms that are able to withstand a wide variety of falsification attacks particularly, those that are launched by a group of colluding malicious routers.

In [1], the authors describe a logarithmic keying protocol for achieving authentication in communication networks. In this scheme, a sender maintains $2\log N$ keys for a set of N receivers. The sender assigns a unique $\log N$ bit identifier to each receiver and gives each receiver a unique subset of $\log N$ keys using the bit-values in the receiver's identifier. The authentication technique is that, the sender signs each message with all its keys and each receiver verifies those signatures for which it has the corresponding signing key. Since, by construction, each receiver maintains a different subset of the sender keys it is not possible for any single receiver to replace all the sender signatures. However, in this scheme, a select pair of users i.e., users who have complementary binary identifiers can collude and compromise all the keys of the sender.

In this work, we describe a family of key distribution protocols that use the same authentication techniques from [1]. We show that the collusion resistance of our protocols depends on the number of keys stored by the sender. For practical purposes, we assume that the sender storage is directly proportional to the communication network diameter and hence, is logarithmic in size. The network diameter assumption can be justified by the fact that most well-known network topologies including the class of hypercubic graphs [5], random graphs in the $G(n, p)$ model with $p = O(1/\log n)$, the Internet, and most peer-to-peer networks [6] have a small diameter.

Threat Model and Assumptions. Our threat model considers a group of malicious nodes that can falsify data and impersonate as the sender. We assume a communication network where critical data is exchanged among the nodes for network related tasks. We do not assume any support from the underlying network other than best effort message delivery. A node can be a router or an end-host and is capable of symmetric key signature generation and verification. We assume that the nodes follow the semantics of communication i.e., the nodes do not drop packets or launch denial of service attacks. These problems are orthogonal to the issues addressed in this current work.

Related Work. In [2], the authors describe a collaborative shared key technique for authentication which has low storage and signature cost. However, it can be shown that this technique can compromised by an arbitrary number of routers. In [4], the authors propose a scheme where the number of keys stored by users is equivalent to the probability of compromise. The only drawback in this scheme is that a large number of keys need to be stored by users. In [3, 7], the authors describe one-way hash chain based techniques for message authentication. In [3], the protocol requires loose synchronization of routers which cannot be guaranteed in most real-time applications. In [7], the authentication approach requires complex setup and protocol state overhead making it more expensive than the protocol from [1]. In the next section, we describe our protocols which improve upon the collusion resistance properties of the protocol described in [1].

3 Our Approach

We describe our family of key distribution protocols in Section 3.1 and give a detailed analysis of their collusion resistance in Section 3.2.

3.1 Key Distribution Protocols

Initially, the sender assigns a unique kl-bit identifier to each receiver where k and l are chosen according to network parameters. The identifier is viewed as k pieces of length ℓ bits. For each piece, a sender maintains 2^{ℓ} keys, one key for each possible value of ℓ bits. A key is represented as $K_i^{b_1 b_2 \cdots b_\ell}$ for the ith piece, $1 \leq i \leq k$ and $b_j \in \{0, 1\}$ for $1 \leq j \leq \ell$. The values of k and ℓ are chosen depending on the desired collusion resistance or the corresponding network diameter.

Now, if the identifier of a receiver t is $t_1 t_2 \cdots t_k$ then, $t_1, t_2 \ldots t_k$ are each l-bit values. For each of these t_i's in the receiver's identifier, the sender gives the receiver the key that corresponds to the l-bit value from the 2^{ℓ} keys that it stored in i^{th} position. For example, if $t_1 = 001$, then the sender gives the receiver the key, K_1^{001} from the $K_1^{2^{\ell}}$ keys that are associated with the first identifier location. Thus, in this key distribution protocol, the user stores k keys per-sender and the sender stores $k2^{\ell}$ keys. When $k = \log N$ and $l = 1$, then, the above protocol is the same that is described in [1]. Furthermore, for different values of k and l we get a different member from this family of key distribution protocols.

Using our protocols, we achieve authentication using an approach that is similar to that described in [1]. We describe the authentication approach for unicast and multicast communication. For unicast, the sender computes an XOR of the all keys held by the user and uses this combined secret to compute the message authentication code, e.g., by using SHA-1, for the message. For multicast, the sender computes several message authentication codes using each of the keys that are in the union of all keys held by the multicast receivers. A user verifies those signatures for which it has the corresponding signing keys to detect message tampering. In the next section, we analyze the collusion resistance of these protocols and show that they perform well in most practical scenarios.

3.2 Collusion Resistance of Our Protocols

Suppose that a message M from source s to destination t is traversing a path P with $u_1, u_2, \cdots, u_d$ as the intermediate nodes in the path. We wish to guarantee that M cannot be tampered with even if all the d intermediate nodes collude. For this we will compute the desired value of ℓ. We use K_{st} to denote the set of keys used by a sender s to sign the message for a receiver t.

Let X_i denote the random variable that the intermediate node u_i can tamper the signature with respect to key K_{st}^j where j denotes the j^{th} key from K_{st}. From our key distribution protocol described in Section 3.1, we have: $E[X_i] = 1/2^{\ell}$.

Now, let $X = \sum_{i=1}^{d} X_i$, then, X denotes the random variable that any of the d intermediate nodes have success in tampering with the signature of key K_{st}^j. Assuming independence, it also holds that using Chernoff bounds, that, $E[X \geq 1] \leq e^{\frac{-2^{\ell}}{2d}}$. Let E_π be the event that all the signatures can be tampered along a path π. Then, $\Pr[E_\pi] \leq \exp\{-\frac{2^{\ell}k}{2d}\}$.

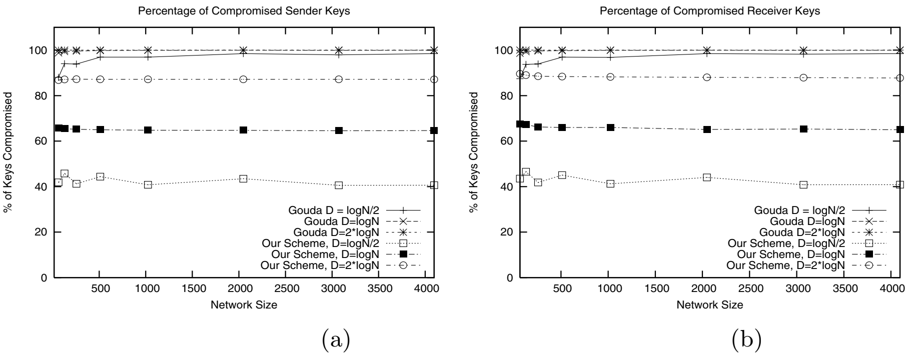
Let E be the event that there exists some path of length d over signatures can be tampered. The number of paths of length d can be at most n^{d+1} since any node of the path can be chosen in n ways. Thus, applying the union bound of probability we have: $\Pr[E] = \Pr[\cup_\pi E_\pi] \leq \sum_\pi \Pr[E_\pi] \leq n^{d+1} \cdot \exp\{-\frac{2^\ell k}{2d}\}$.

Choosing ℓ and k: From the above analysis, the value of ℓ and k depend on d. A cursory glance reveals that in fact d should be the diameter of the network. Considering logarithmic-diameter networks, for the event E to have a low probability, we require that $n^{d+1} \cdot \exp\{-\frac{2^\ell k}{2d}\} \leq n^{-c}$ for some constant $c > 1$. Simplifying, we require that $(d+1)\log n - 2^\ell k/2d < -c \log n$ or $2^\ell k > O(d^2 \log n)$. By letting $k = O(\log n)$, we can choose $\ell = O(\log d)$. With these values of k and ℓ, it now implies that for an n-node network, nodes have to choose identifiers of length at most $O(\log n \log d)$ bits.

4 Results

We analyze the performance of our proposed family of key distribution protocols with respect to the protocol from [1]. We show that the collusion resistance in our protocols, for most practical networks with a given diameter, is better than the protocol from [1]. Our results aid a system designer in choosing an appropriate protocol based on the network diameter and/or the probability of collusion among nodes.

Our experiments were simulated on random network sizes with different diameters and averaged over 1000 trials. In Figure 1, we show the percentage of the signatures that are compromised for networks with diameters: $\log N/2$, $\log N$ and $2\log N$ for a total of N users. This percentage represents the number of forged signatures that can be generated by the colluding users. For this comparison, we chose the value of $2^l = \log N$, i.e., the number of keys stored by the sender as $\log^2 N$. We compare the performance of our scheme (termed, "Our Scheme") with the scheme from [1] (termed, "Gouda"). In Figure 1(a), the number of sender signatures that are compromised along a diameter are shown. We see that our protocol exhibits better collusion resistance even when the network diameter is as large as $2\log N$. In Figure 1(b), we show the scenario where the keys held by an individual receiver are compromised by nodes that are along the



(a) (b)

Fig. 1. Percentage of Keys Compromised for, (a) Sender (b) Receiver

path from the sender to the receiver. We observe that, even for large network diameters, the receiver still has some keys that are not compromised and hence, signature verification by the receiver can detect falsification attempts. Hence, these results show that for most practical situations our protocols provide the necessary collusion resistance.

We have compared the collusion resistance of different members of our family of protocols. Our results show that, by progressively choosing smaller/larger values of 2^l we are able to trade off collusion resistance with cost of signing the messages. Due to lack of space we do not present those results.

5 Conclusions

The paper described a new model of session-level connectivity provisioning for use by QoS-sensitive networked applications. The connectivity provider (SP) employs policy functions to map the application-prescribed flow specs onto the bandwidth needs of connections that carry the data flows. The strategy is to reduce the per-flow cost incurred by multiplexing many closely-similar data flows on a single connection. The multiplexing brings in two benefits to the SP. First, it reduces the per-flow bandwidth allocation due to the gains accrued from a statistical sharing of connection bandwidth. Second, it amortizes the connection-level overhead across many flows. The level of cost reduction, and hence the revenue accrual, can be controlled by the SP using policy functions that take into account the burstiness and loss/delay tolerance of data flows.

Our studies indicate that the model can be employed in large network settings (such as IP networks), while dealing with the scalability issues arising therein.

References

[1] Gouda, M.G., Kulkarni, S.S., Elmallah, S.E.: Logarithmic keying of communication networks. In: 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS-06 (2006)

[2] Huang, D., Sinha, A., Medhi, D.: A double authentication scheme to detect impersonation attack in link state routing protocols. In: IEEE International Conference on Communications, vol. 3, pp. 1723–1727 (2003)

[3] Perrig, A., Canetti, R., Tygar, D., Song, D.: The TESLA broadcast authentication protocol. Cryptobytes 5(2) (2002)

[4] Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., Pinkas, B.: Multicast security: A taxonomy and some efficient constructions. In: IEEE INFOCOMM (1999)

[5] Scheideler, C.: Universal Routing Strategies for Interconnection Networks. LNCS, vol. 1390. Springer, Heidelberg (1998)

[6] Bhargava, A., Kothapalli, K., Riley, C., Thober, M., Scheideler, C.: Pagoda: An overlay network for data management, routing and multicasting. In: ACM SPAA, pp. 170–179 (2004)

[7] Hu, Y.-C., Perrig, A., Johnson, D.: Efficient security mechanisms for routing protocols. In: NDSS. Network and Distributed System Security Symposium (2003)