Toward Secure Key Distribution in Truly Ad-Hoc Networks

Aram Khalili
Jonathan Katz
William A. Arbaugh
{aram,jkatz,waa}@cs.umd.edu

Department of Computer Science
University of Maryland (College Park)

Abstract

Ad-hoc networks — and in particular wireless mobile ad-hoc networks — have unique characteristics and constraints that make traditional cryptographic mechanisms and assumptions inappropriate. In particular, it may not be warranted to assume pre-existing shared secrets between members of the network or the presence of a common PKI. Thus, the issue of key distribution in ad-hoc networks represents an important problem. Unfortunately, this issue has been largely ignored; as an example, most protocols for secure ad-hoc routing assume that key distribution has already taken place.

Traditional key distribution schemes either do not apply in an ad-hoc scenario or are not efficient enough for small, resource-constrained devices. We propose to combine efficient techniques from identity-based (ID-based) and threshold cryptography to provide a mechanism that enables flexible and efficient key distribution while respecting the constraints of ad-hoc networks. We also discuss the available mechanisms and their suitability for the proposed task.

1. Introduction

Personal wireless devices are becoming increasingly common and are expected to become ubiquitous in certain domains. These mobile devices can form ad-hoc networks without any supporting infrastructure or prior organization. With the proliferation of portable devices, the availability of security services for these networks becomes more important. Our goal is to provide security mechanisms that can support diverse applications and security policies in ad-hoc networks while making as few assumptions as possible about the nature of the network or the application.

A crucial difference between ad-hoc networks and traditional networks is the lack of central administration or control over the network principals. This is problematic for security mechanisms because in traditional networks the central administration often defines the security services and policies for the network, and may also pre-distribute keys to all participants. Since the standard methods for providing keys (e.g., using a centralized administrator) are unavailable, keys must be dynamically generated *by the network itself* at the time of network formation. Furthermore, new participants joining the network must be able to obtain keys — and distribute the corresponding public keys throughout the rest of the network — in an ad-hoc manner.

We stress that we explicitly avoid assumptions that consider the key distribution problem solved. In many ad-hoc environments, it is simply incorrect to assume that pairwise secrets exist between devices in the network, or to assume that every principal has a public-key certificate which is verifiable by all other principals (indeed, principals may not share the same certification authority). Our goal, then, is to provide a protocol for distributing keys and establishing an underlying infrastructure on top of which other protocols (e.g., protocols for secure ad-hoc routing) can be run. Note that most security protocol for ad-hoc networks require some prior distribution of keys.

1.1. Ad-hoc network constraints

We envision ad-hoc networks to be formed by nodes without any prior contact, trust, or authority relation. This precludes any pre-distributed symmetric keys or a reliable (external) PKI supported by all nodes. We assume that all nodes are resource-constrained in energy, bandwidth, computational ability, memory, and possibly long term storage as well. We pay particular attention to energy utilization, since this is often the most severe constraint. Also, the energy required to transmit/receive messages in a wireless network can be equivalent to the energy used by several thousand cycles of the CPU.[1] When sending messages, the energy consumption arises from the need to transmit a sufficiently powerful signal for good signal/noise ratio, while for receiving messages, the energy consumption comes from the signal processing necessary to decode a spread-spectrum signal.

We also assume that nodes are mobile and that due to this and other environmental conditions the topology of the network can change frequently; thus, some nodes may be unreachable for some of the time. The nodes are also assumed to have low physical security; i.e., we assume they can easily stolen or otherwise compromised by an adversary. Thus our adversarial model includes active (or Byzantine) adversaries who can compromise some bounded fraction of nodes in the network.

Finally, we assume that fewer than $1/3$ of the principals at the time of network formation are corrupted or malicious. Our solutions can be modified to tolerate corruption of up to half of the participants under certain physical assumptions about the network (e.g., that all nodes at the time of network formation share a broadcast channel). We note that when even stronger assumptions can be made (i.e., that no malicious nodes are present at the time of network formation) the efficiency of our solution can be greatly improved.

1.2. The current state of ad-hoc network security

Ad-hoc network security research often focuses on secure routing protocols, which form an essential component of security in ad-hoc networks [10]. However, all such routing schemes known to us neglect the other crucial challenge in ad-hoc security: key establishment and distribution. Protocols such as ARAN, Ariadne, SEAD, SPINS, and SRP [6, 9, 8, 15, 14] all assume the pre-existence and pre-sharing of secret and/or public keys for all (honest) principals.

This leaves ad-hoc key management and key distribution as a wide open problem. Intuitively, this should not be very surprising, as the distribution of keys in networks often mirrors trust (or authority) relations in the real world, and ad-hoc networks may not have any pre-existing trust relations. A new mechanism is needed that can accommodate the new trust scenarios in ad-hoc networks. Only recently have approaches for key distribution in ad-hoc networks been proposed [16, 1].

Zhou and Haas [16] introduce the idea of distributing a certificate authority (CA) throughout the network, in a threshold fashion, at the time of network formation. This CA would allow trust relations to be created in the network while also being resilient to some intrusions, malicious insiders, and breaks in connectivity. However, Zhou and Haas do not address the resource limitations of devices in ad-hoc networks. Public-key and threshold cryptography are (in general) computationally expensive and need to be tailored to the resources and constraints of low-power devices. A key management and distribution scheme that is efficient enough to be feasible for resource-constrained devices can provide the infrastructure needed by protocols for secure ad-hoc routing and can therefore enhance the set of services available for securing ad-hoc networks.

[1] See, e.g., `http://xbow.com`.

2. Proposed ad-hoc keying mechanism

In this Section, we briefly outline our approach which combines the ideas of ID-based and threshold cryptography; we then review each of these components. Our suggested solution is described in more detail in Section 3, and we recommend a particular instantiation in Section 4.

We propose using a threshold, ID-based cryptosystem to achieve security, efficiency, and resilience. Instead of assuming that prior keying material or trust/security associations exist, we establish these at the time of network formation. In more detail, we propose that (at the time of network formation) the participating nodes generate — in a distributed fashion — a master public key PK^* for an ID-based cryptosystem. The master secret key SK^* will be shared in a t-out-of-n threshold manner by this initial set of n nodes. Having established PK^*, all principals in the network — including those joining at later times — can now use their *identity* as their public key (the master public key will be distributed throughout the network and known by everyone). Of course, participants will have to obtain the secret key corresponding to their identity; this key will be computed by having the participant obtain t shares of their key (after appropriate authentication) from t-out-of-n of the original nodes in the network. Note that distributing the key generation service in a t-out-of-n fashion requires an adversary to corrupt at least t nodes in order to learn a user's secret key. Furthermore, honest parties need only contact *any* t nodes in order to obtain their own key, thus making the protocol resilient to temporary loss of connectivity with other nodes in the network.

Our suggestion is similar in spirit to that of Zhou and Haas [16] (we replace their threshold CA by a threshold private-key generation service). However, we note the following advantages of our approach:

- We avoid the need for users to generate their own public keys and to then distribute these keys throughout the network. In our setting, the user's identity acts as their public key. This significantly reduces the computation necessary to join the network.

- In a CA-based solution, a user is required to propagate both his public key as well as a signature (by the CA) on his public key. In an ID-based system, users need only propagate their identity (which is typically included in every message anyway). This can lead to huge savings in bandwidth.

- For particular ID-based systems, we expect that initial network formation (i.e., distributed key generation) will be more efficient than distributed key generation for a threshold signature scheme (e.g., an RSA-based signature scheme).

Note that we do not specify the nature of the identity to be used, nor do we specify a means for authenticating users' identities before sending them (shares of) their secret key. We suggest some possibilities in Section 3.

2.1. ID-based cryptography

We provide a high-level overview of ID-based *encryption* here, and refer the reader elsewhere [3] for details. We mention that ID-based signature schemes are also known.

In an ID-based encryption scheme, a master public key/secret key is generated by a private-key generation service (PKG) and the master public key is assumed to be known by everyone. Once this master public key is established, *arbitrary identities* may be used as public keys for the scheme. In other words, a sender can encrypt a message for a recipient with identity ID using only the master public key and the string "ID"; in particular, the recipient ID does *not* have to establish his own public key and propagate it throughout the network. In order to decrypt a message encrypted under a particular identity ID, the user ID must obtain the secret key corresponding to his identity from the PKG. Only the PKG (who knows the master secret key) can generate personal secret keys for various identities.

Roughly speaking (see [3] for more details), encryption of messages for a particular user ID remains secure as long as an eavesdropper does not have the master secret key nor the personal secret key corresponding to ID. Thus, in particular, obtaining (multiple) personal secret keys for identities *other than* ID does not help an adversary break the security of the scheme. This is crucial since the adversary may be a party in the network and may therefore obtain personal secret keys for identities other than ID from the PKG.

In more detail, an ID-based encryption scheme consists of the following algorithms:

`Setup` takes as input a security parameter and returns the master public/secret keys for the system.

`Extract` takes as input the master secret key and an identity (which is an arbitrary string) and returns the personal secret key corresponding to the identity.

`Encrypt` takes as input the master public key, the identity of the recipient, and a message and returns a ciphertext.

`Decrypt` takes as input the master public key, a ciphertext, and a personal secret key and returns the plaintext.

We stress that ciphertexts are strongly tied to a particular identity; i.e., a ciphertext C intended for ID will result in gibberish if some $ID' \neq ID$ attempts to decrypt C using his own personal secret key.

2.2. Threshold cryptography

Again, we provide a high-level overview only and refer the reader elsewhere (e.g., [7]) for details.

Threshold cryptography allows a cryptographic operation to be "split" among multiple users such that only some threshold of the users can perform the desired operation. In a t-out-of-n threshold scheme, any set of t users (out of a total of n users) can compute the desired functionality while any set of $t-1$ users cannot. In particular, the implies that (1) an adversary who compromises $t-1$ users cannot compute the desired functionality (and the scheme remains secure against such an adversary); furthermore (2) an honest user who needs the cryptographic operation to be performed need only contact (any) t of the users. We note that threshold scheme secure against byzantine adversaries exist for $t < n/2$ only (when a broadcast channel is assumed to exist); schemes secure against passive adversaries can support $t < n$.

Although we do not specifically deal with the issue here, we note that proactive threshold schemes [13] can tolerate an adversary who corrupts even *all* the users in the network, so long as no more than t users are corrupted at any given time. It is usually assumed that corrupted nodes will be detected eventually, and these nodes will be "rebooted" so as to return to honest behavior.

We note also that distributed schemes supporting other (i.e., non-threshold) access structures are possible, but these schemes are typically less efficient than threshold schemes. Furthermore, in an ad-hoc environment there may be no *a priori* reason to distinguish between the different nodes, and hence no reason to deviate from a threshold access structure.

3. A combined approach to ad-hoc keying

We now describe our proposed solution in more detail. At the time of network formation, the nodes that are forming the network decide on a mutually acceptable set of security parameters. Any node which is not satisfied by the choice of parameters can choose to refuse to participate in the network. The security parameters might include a threshold t of key service nodes, the number and identity of key service nodes, particular parameters of underlying schemes (e.g., key lengths), and a policy for key issuance. This initial negotiation is independent of our proposed scheme and is not discussed in any detail. It should be noted, though, that the initial policy negotiation is a potential target for active or byzantine adversaries, and the negotiation protocol should address this issue.

This initial set of nodes can then form a threshold PKG for an ID-based scheme. These nodes will generate the master secret/public keys in a distributed manner such that fewer than t nodes cannot recover the master secret key. The master public key is given to all members of the network when they join, and the PKG can start issuing personal secret keys to nodes (including themselves) based on their identities and the key issuance policy. An identity can be something usually present in transmitted messages, like a MAC (or other network layer) address. To receive the private key corresponding to some identity, a node presents this identity and any extra material specified by the key issuance policy to t (or more) nodes forming the PKG and receives a share of their personal private key from each of them. With t correct shares, the node can then compute its personal private key within the network's ID-based system. An efficient local mechanism is provided to check the correctness of the individual shares and the computed private key.

Distributing the key generation and the PKG service prevents a single point of failure and resists compromise or insider attack (up to the threshold k). Also, distributing the PKG in a t-out-of-n fashion makes the scheme resilient when some nodes are unreachable due to ad-hoc conditions as long as at least k are still reachable. It is also possible that, at times, there will not be k members of the PKG within communication range. (We cannot necessarily rely on secure routing to communicate with at least k nodes since key associations are not yet set up!) However, we suggest to use mobility to one's advantage: if there are fewer than k members of the PKG who are reachable, a node can obtain some number of shares of his key, and then move to try to discover more key service nodes in order to obtain a total of k shares. (Note that members of the PKG do not need to communicate when handing out shares of personal private keys.)

We stress that our scheme makes no assumption about the "security" of users' identities, e.g., that they are set in hardware or cannot be spoofed. The key issuance policy needs to address this, however. Our scheme does "localize" the problem in that spoofing only needs to be prevented/detected by the nodes forming the PKG at the time of key issuance (and this can be done by requiring some "unspoofable" supporting material to be presented at the time of a key request); spoofing need not be a concern for other nodes in the network at other times.

For completeness, we recommend a number of possibilities for user identities. One possibility is to use statistically unique cryptographically verifiable (SUCV) addresses [12] (applied to ad-hoc networks by [1]). However, a simpler option is to assume that identities are *unpredictable* (which would be the case, e.g., if identities are randomly chosen by nodes joining the network). In either case, members of the PKG should also refuse to issue keys for a particular identity more than once. Note that this effectively solves the spoofing problem: because identities are unpredictable, an adversary will be unable to obtain someone's personal private key in advance; furthermore, since keys are not reissued the adversary will be unable to obtain a node's personal key once that node has already obtained it.

A CA-based solution [16] requires transmission, storage, and verification of public keys and certificates, where the size of each of these (e.g., in an RSA- or El Gamal-based implementation) is the key length of the underlying cryptographic scheme. ID-based systems, on the other hand, avoid the need for this: the user's identity (and the master public key) is sufficient to derive the corresponding "public key", and no certification is necessary. This can lead to huge savings in bandwidth. As an example, RSA keys are typically ≥ 1024 bits, hence using an ID-based scheme saves more than an order of magnitude in communication (an RSA public key plus the CA's signature thereon requires 2048 bits; an identity can be 64–128 bits long). We also save in computation, as no verification of a certificate is necessary (because ID's serve as public keys). The ID-based schemes we mention below (which are based on elliptic-curve cryptography) have very short ciphertexts/signatures and efficient computation times. Finally, we suggest a particular instantiation for the ID-based scheme which leads to very efficient distributed key generation (more efficient than in the case of RSA-based systems).

4. Efficient ID-based and threshold schemes

ID-based cryptography is relatively new field with few existing schemes. Although the Maurer-Yacobi scheme [11] has been suggested as an ID-based scheme, it is unsuitable for our purposes as it does not achieve provable security in an adversarial model appropriate for our setting. The Boneh-Franklin scheme [3] (the first provably-secure ID-based encryption scheme) seems quite suitable. It is based on elliptic-curve cryptography, giving savings in computation and communication. The ciphertext of a (short) message block in the Boneh-Franklin scheme is 2–4 times the size of the plaintext, depending on whether chosen-ciphertext security is desired. In either case (of course), the scheme can be used to encrypt a (short) symmetric key that is used to encrypt an arbitrarily-long message; this results in only a small constant overhead in communication. For ID-based signatures, we recommend, e.g., the scheme of Cha and Cheon [5] which (building on work of Boneh, Lynn, and Shacham [4]) yields signatures that are both very efficient to compute and extremely short (roughly 160 bits long). We note that both the Boneh-Franklin and the Cha-Cheon schemes use similar elliptic-curve groups, and hence can be combined (for greater efficiency) in a relatively straightforward manner.

Both the ID-based schemes mentioned above use so-called Gap Diffie-Hellman (GDH) groups; i.e., groups in which the decisional Diffie-Hellman problem is efficiently computable, but the computational Diffie-Hellman problem is assumed to be hard. Boldyreva [2] shows that this structure can also be exploited in distributing such schemes efficiently. In particular, it allows the usual zero-knowledge interactive proofs for share verification to be replaced by (local) DDH computations. Boldyreva uses a distributed key generation algorithm due to Gennaro, et al. [7], which tolerates up to $\lfloor \frac{n}{2} \rfloor$ malicious nodes. This is optimal for threshold scheme, but comes at an efficiency cost. If simplifying assumptions can be made (e.g., that there are no malicious nodes at the time of network formation), the efficiency of the master key generation for the ID-based scheme can be improved. It may also be possible to exploit the algebraic structure of GDH groups for further computational improvements in key generation.

5. Summary

Ad-hoc networks cannot always be assumed to have keying material or mechanisms for key distribution in place at network formation time. We suggest a mechanism that allows creation of a keying service in the network; this service is efficient, robust, and respects constraints and characteristics of ad-hoc networks. We propose to do this by a novel combination of two cryptographic techniques: ID-based and threshold cryptography. ID-based cryptography primarily provides efficiency gains, and threshold cryptography provides resilience and robustness. We have identified particular schemes as candidates for implementing our approach.

We stress that we do not claim that our solution completely handles the issue of key distribution in ad-hoc networks. For example, our scheme is vulnerable to main-in-the-middle attacks on joining members. Malicious members of the network can also provide newly-joining members with a false master public key, perhaps one for which the malicious member holds the corresponding master secret key. Much work remains to be done both to formalize a model of security for ad-hoc networks, and to present provably-secure solutions in this model.

References

[1] R. B. Bobba, L. Eschenauer, V. Gligor, and W. A. Arbaugh. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. Technical Report, Institute for Systems Research, UMd, TR 2002-44, 2002.

[2] A. Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In *International Workshop on Practice and Theory in Public Key Cryptography*, January 2003.

[3] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Killian, editor, *Advances in Cryptology, CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer Verlag, August 2001.

[4] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Advances in Cryptology, ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 512–532. Springer Verlag, 2001.

[5] J. C. Cha and J. H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In *International Workshop on Practice and Theory in Public Key Cryptography*, January 2003.

[6] B. Dahill, B. Levine, E. Royer, and C. Shields. A Secure Routing Protocol for Ad Hoc Networks. Technical Report UM-CS-2001-037, University of Massachusetts, August 2001.

[7] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Eurocrypt, 1999.

[8] Y.-C. Hu, D. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *Workshop on Mobile Computing Systems and Applications*. IEEE, June 2002.

[9] Y.-C. Hu, D. B. Johnson, and A. Perrig. Secure On-Demand Routing Protocols in Ad Hoc Networks. Unpublished, 2001.

[10] A. Khalili and W. A. Arbaugh. Security of wireless ad-hoc networks. Work in progress, `http://www.cs.umd.edu/~aram/wireless/survey.pdf`, 2002.

[11] U. Maurer and Y. Yacobi. Non-interactive public key cryptography. In *Advances in Cryptology, EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 489–507. Springer Verlag, August 1991.

[12] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses. In *Proceedings of the 2002 Networks and Distributed Systems Security conference*, February 2002.

[13] R. Ostrovsky and M. Yung. How to Withstand Mobile Virus Attacks. PODC, 1991.

[14] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad hoc Networks. In *Communication Networks and Distributed Systems Modeling and Simulation Conference*, January 2002.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *Proceedings of Mobile Computing and Networking*, 2001.

[16] L. Zhou and Z. Haas. Securing Ad Hoc Networks. *IEEE Network Magazine*, 13(6), November/December 1999.