Provable security for ad hoc routing protocols

GERGELY ÁCS, LEVENTE BUTTYÁN, ISTVÁN VAJDA

Budapest University of Technology and Economics, Department of Telecommunications
Laboratory of Cryptography and Systems Security (CrySyS)

{acs, buttyan, vajda}@crysys.hu

Reviewed

Key words: ad hoc networks, on-demand ad hoc source routing, secure ad hoc routing, provable security, simulation paradigm

In this article we present a new formal framework that can be used for analyzing the security of on-demand source routing protocols proposed for wireless mobile ad hoc networks. Our approach is based on the simulation paradigm which is a well-known and general procedure to prove the security of cryptographic protocols. We give the formal definition of secure ad hoc routing in a precise and rigorous manner using the concept of statistical indistinguishability. We present an ad hoc source routing protocol, called endairA, and we illustrate the usage of our approach by proving that this protocol is secure in our model.

1. Introduction

An ad hoc network is the cooperative engagement of a collection of wireless mobile nodes without the required intervention of any centralized access point or existing infrastructure. The nodes have terminal and network functions as well. They are often equiped by constrained energy supply (battery). Due to this fact and to reduce the interference of radio communication, the nodes communicate in a multi hop manner. In addition, due to the lack of a pre-deployed infrastructure, all nodes must perform routing and maintenance functions as well.

There exist two sorts of ad hoc routing: pro-active and reactive (or on-demand) protocols. In the rest of the article we deal with the latter one. Considering reactive routing, the source node initiates a route discovery towards a target node only if it needs to communicate with the target. In that case, the initiator node floods the whole network with route request messages *(rreq)*. Every node receiving the request appends its own indentifier to the node list that is placed in the request message and re-broadcasts the message. When the target node receives the route request it replies with one or more route reply messages *(rrep)* that contain the node list received in the request message. This node list itself is the discovered route. The reply travels back to the source node on the reverse of the route carried by the request.

Secure ad hoc routing means that the correct operation of the above mechanism is ensured even in the presence of an adversary. This has primary importance, since by manipulating the route discovery process, an adversary can paralyse the entire network using relatively small amount of resources.

Several “secure” ad hoc routing protocol have been proposed so far ([3] gives a deep overview of this topic), however the authors of these protocols have not proved their proposals by formal means. To the best of our knowledge, [2] is the first work that contains a precise mathematical model that is applicable for analyzing the security of ad hoc routing protocols. In [2], the authors present new subtle attacks against two well-known protocols (SRP, Ariadne) and at the same time they propose a new routing protocol that is provably secure in that model. The model that was used in [2] is based on the simulation paradigm which is a well-known method to give a formal proof of the security of various cryptographic protocols [4,5]. However, that model assumes a constrained Active-1-1 adversary that controls only one compromised device and uses only one compromised identifier. A further restriction is that the adversary can attack the execution of only one route discovery process.

In the present article, we generalize the model used in [2] for the case of an Active-y-x adversary and parallel execution of several insatnces of the routing protocol. Further, we show that endairA is secure in this extended model too. Due to space limitations, here we can only describe the basics of our approach; one can read about the complete work in [1].

2. Formal model

2.1. Modeling the network

We consider static ad hoc networks that are modeled by undirected labeled graphs $G(E,V)$, where each vertex uniquely corresponds to a node and there is an edge between two vertices if and only if the corresponding nodes can overhear each others communication (i.e., they are neighboring nodes). We assume that each node has an identifier that identifies the node unambiguously (e.g. a public key if we use public key cryptography). We further assume that all of the identifiers are authenticated but some of them are compromised by an adversary and the keys that are needed for their authentication are possessed by the adversary. An ad hoc network together with an adversary is shortly called a *configuration*. Formally a configuration is a triplet $G(E,V)$, $V^*, L)$, where $G(E,V)$ is a graph representing the network, $V^* \subset V$ is the set of nodes that are controlled by

the adversary, and *L* is a labelling function that assigns to each vertex the set of identifiers that belong to the node corresponding to this vertex. This set of identifiers is a singleton in case of honest nodes, but all of the corrupted identifiers are assigned to every corrupted node.

2.2. Modeling the adversary

We make the following assumptions about the adversary:

- the advesary cannot be physically present everywhere at the same time, thus it is not able to control the entire network;
- the adversay controls *x* nodes and uses *y* compromised identifiers (Active-*y*-*x* adversary, where $x, y \geq 1$);
- the set of compromised identifiers used by the adversary and the set of identifiers used by the honest nodes are disjoint;
- a corrupted node has the same communication capabilities as the honest nodes meaning that each malicous node can send messages only to its neighbors and they can overhear only the communication of neighboring nodes.
- the adversary is active in the sense of that besides eavesdropping messages, it can fabricate and insert new messages, and in addition it can modify and delay existing messages;
- the adversary is not adaptive meaning that it cannot coerce honest nodes to start route discoveries based on the information that it obtained in previously initiated route discoveries.

We assume that the initiator and the target of every route discovery process are honest.

2.3. Definition of plausible route

It is not a trivial task to give a formal definition of secure routing. The requirement of returning the most optimal (in some cases shortest) path seems to be a simple solution to this problem, however due to the varying message delaying and the applied optimizations, it seems to be an unrealistic requirement [2]. Further one can see that we cannot prevent a corrupted node from inserting arbitrary corrupted identifiers (even many times) into the node list carried by an intercepted routing message, and similarly we are not able to prevent the neighboring corrupted nodes from exchanging any information freely [1]. Now it should be clear that there are some attacks in practice which are unavoidable or it is very costly to defend against. Consequently, we have to form the notion of security with care: if we give a too strong definiton, then due to the unavoidable attacks mentioned above, no routing protocol will satisfy our definiton, and on the other hand if our definiton is too weak, then there will be protocols that are secure in our model but could be vulnerable to various attacks besides the unavoidable ones.

We solve this problem by embedding the possibility of unavoidable attacks in the definition of "correct routes". We call the routes that satisfy our definition of correctness *plausible* routes. The formal definition is given below. Every configuration can be unambiguously reduced to another configuration that has a graph without neighboring corrupted nodes. In other words we merge the neighboring vertices that correspond to neighboring corrupted nodes into a single one in the reduced graph. We denote this graph by $\underline{G}$.

Definition:

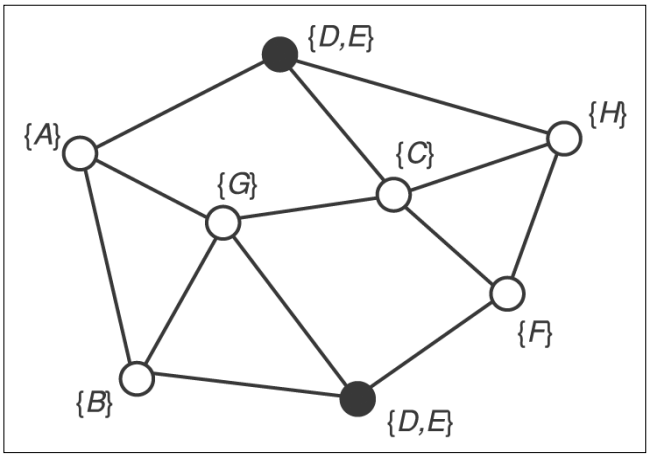
A sequence of identifiers is a plausible route, if

– it does not contain any repeating identifiers and

– it can be partitioned into sub-sequences in such a way that each of the resulting partitions is a subset of the identifiers assigned to a vertex in $\underline{G}$, and in addition, these vertices form a path in $\underline{G}$.

A reduced configuration is depicted on *Figure 1.* for illustration purposes. The solid verticies are representing the merged corrupted nodes that use compromised identifiers *D* and *E*. It is easy to see that *A,D,E,C,F* is a plausible route and a correct partitioning of this route is *A|D,E|C|F*, but *A,B,D,E,H* is not a plausible route, since the nodes that correspond to identifiers *E* and *H* are not neighboring.

Figure 1.



2.4. Simulation paradigm

The pivot of a formal model is to precisely define what we mean by secure routing. To achieve this goal we would like to apply the widely used simulation paradigm [4,5].

The fundamental idea of the simulation paradigm is that the adversary gains nothing if whatever it can achieve by unconstrained adversarial behaviour can also be achieved within essentially the same computational effort by a benign behaviour. The definition of the benign behaviour captures what we want to achieve in terms of security, and in our case, it is realted to the concept of plausible routes. In this model, we define a real-world model, that describes the real operation of the protocol under investigation, and the ideal-world one represents the ideal operation of this protocol. One can think of the real-world model as an implementation of the protocol, while the ideal-world model can be considered as a specification. Both models contain adversaries. The ideal-world adversary represents the

unavoidable attacks, or in other words, the tolerable imperfections of the system. On the other hand, we do not constrain the real-world adversary, but we assume that it can perform any polynomial-time attacks in the security parameter and in the size of the network.

A protocol is secure if for any unconstrained real-world adversary *A* there exist an ideal-world adversary *A'* such that *A* gains nothing more substantial than *A'* using the same computational effort. In other words, the behaviour of *A* can be simulated by the behaviour of *A'*, in the sense that the outputs of the ideal- and real-world models are indistinguishable from the point of view of the honest protocol participants. Intuitively, if any real-world adversary can be simulated by an appropriate ideal-world adversary, then there is no real-world adversary that can perform more than the unavoidable attacks.

In the followings we formally define the ideal- and real-world models of ad hoc routing protocols then we more precisely define the indistinguishability of the two models.

2.5. Real-world model

The real-world model is depicted on *Figure 2.* This model consists of the set of interacting and probabilistic Turing machines that communicate via common tapes. The machines model the operation of the honest protocol participants and the adversary. $M_1,...,M_{n-k}$ represent the honest devices that belong to honest nodes so they correspond to vertices v_i in $V \setminus V^*$.

The corrupted devices are denoted by $A_1,...,A_k$. These corrupted devices belong to corrupted nodes in V^*. Machine *C* models the radio links represented by the edges of $\underline{G}$. The task of machine *C* is to move the protocol messages appearing on the output tapes of the machines to the input tapes of the neighboring machines (neighboring relation is based on $\underline{G}$). Every machine apart from *H* is probabilistic. *H* is an abstraction of higher-layer protocols run by the honest parties meaning that *H* initiates the route discovery procedures by placing request messages on tape req_i. A response to these requests is eventually returned via tape res_i.

Tapes ext_j model an out-of-band channel through which the adversary can instruct the honest parties to initiate route discovery processes from an arbitrary node towards an arbitrary node. Arbitary in this context means that the adversary can choose these nodes. Note that the adversary is non-adaptive, thus it can use these tapes only at the beginning of the computation. So the messages placed on these tapes do not depend on the messages observed by the adversary during the protocol run.

At the beginning of the computation every machine is initialized with some input data (e.g cryptographic keys), which determines its initial state. The probabilistic machines also receive some random input (the coin flips to be used during the operation).When the machines have been initialized, the computation begins. The machines operate in a reactive manner, which means that they need to be activated in order to perform some computation. When a machine is activated, it reads the content of its input tapes, processes the received data, updates its internal state, writes some output on its output tapes, and goes back to sleep (i.e., starts to wait for the next activation). The machines are activated in rounds by a hypothetic scheduler in a specified order. The computation ends when *H* reaches one of its final states.

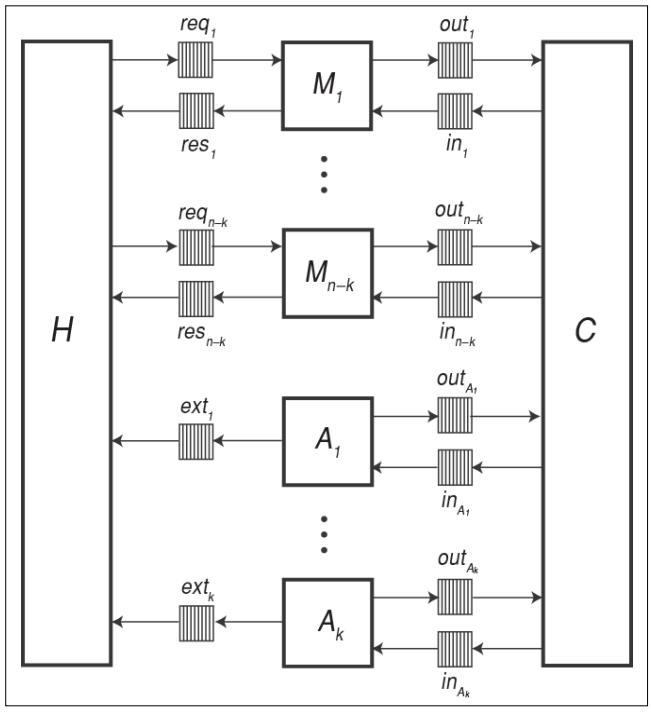
The output of the real-world model is the set of the routes returned to *H*. This output is denoted by $real\_out_{conf,A}(r)$, where *conf* and *A* represent the configuration and the adversary respectively. $r = (r_l, r_{M1}, ..., r_{Mn-k}, r_{A1}, ..., r_{Ak}, r_C)$ is a vector containing the random input of each machine and r_l is the random input used to generate the cryptographic keys. $real\_out_{conf,A}$ denotes the random variable describing the output, when *r* is chosen uniformly at random.

2.6. Ideal-world model

The ideal-world model is shown in *Figure 3.* As one can see, the construction of the ideal-world model is similar to the construction of the real-world model, so here we only describe the differences between them:

- Before machine *C'* places a route reply message *(rrep)* on the tape in_i of a machine M_i, it checks whether the message contains any non-plausible routes. If and only if this is the case, then *C'* puts a corruption flag on the message. Otherwise machine *C'* operates like machine *C*.
- When M_i' receives a route reply message that belongs to a request that was initiated by him, then he performs all of the verifications required by the protocol on the message. If these verifications are successful, then it checks whether the message has a corruption flag. If it has, then M_i' drops the message. Otherwise machine M_i' operates like machine M_i.

Figure 2. The real-world model



The output of the ideal-world model is the set of the routes returned to *H*. This output is denoted by *ideal_out*$_{conf,A}$(r), where the interpretation of *r'* is similar to the interpretation of *r* in the real-world model.

ideal_out$_{conf,A}$ denotes the random variable describing the output, when *r'* is chosen uniformly at random.

It is easy to see that, in the ideal-world model, *H* never receives a route reply message containing any non-plausible routes. In effect, the ideal-world model is ideal in that sense.

2.7. Formal definition of secure ad hoc routing

Considering the comments related to the unavoidable attacks, we require from a secure routing protocol to return non-plausible routes only with negligible probability. We can formally describe this requirement using the two models and the simulation paradigm in the following way:

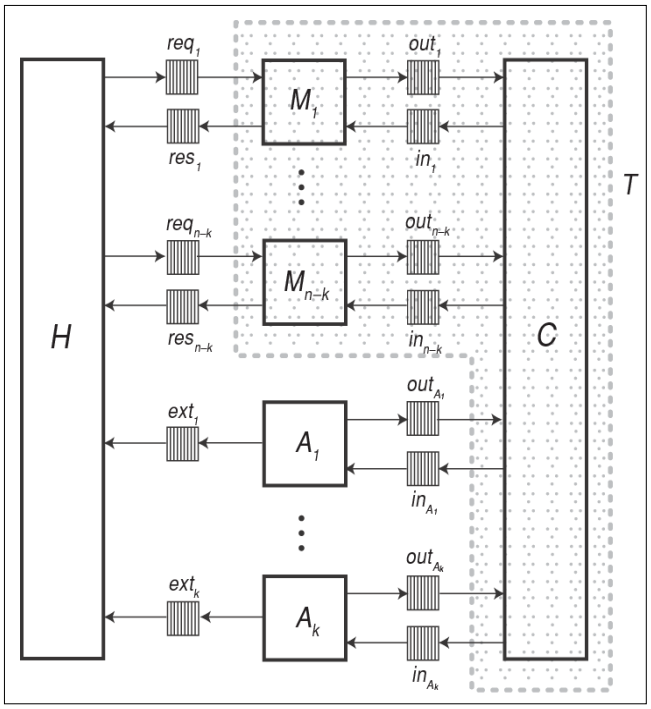
Definition:

A routing protocol is said to be statistically secure if, for any configuration conf and any real-world adversary A, there exists an ideal-world adversary A', such that ideal_out$_{conf,A}$ *is statistically indistinguishable from real_out*$_{conf,A'}$.

In this definition we do not require the exact matching of the distributions, since that requirement could not be satisfied by any protocol in practice. The adversary can always carry out a successful attack against the applied cryptographic primitve with negligible probability (e.g. by forging a correct digital signature).

The above definition can be weakened if we require computational indistinguishability instead of statistical indistinguishability, but in this article we will not need this.

Figure 3. The ideal-world model



3. Security of endairA

In this part we would like to demonstrate the usage of our model by a short example. First, we present a new ad hoc routing protocol, called endairA [2,1], and then we prove that this protocol is secure in our model.

The operation of the protocol is exemplified by the following message exchanges, where sig_x denotes the digital signature of node *x*, and *id* is a non-predictable random request identifier:

Route Request:
S→* : [rreq, S, D, *id*, ()]
B→* : [rreq, S, D, *id*, (B)]
C→* : [rreq, S, D, *id*, (B, C)]

Route Reply:
D→ C : [rrep, S, D, *id*, (B, C), (sig_D)]
C→ B : [rrep, S, D, *id*, (B, C), (sig_D, sig_C)]
B→ S : [rrep, S, D, *id*, (B, C), (sig_D, sig_C, sig_B)]

In endairA, the initiator of the route discovery process generates a route request *(rreq)*, which contains the identifiers of the initiator (**S**) and the target (**D**), and a randomly generated request identifier *(id)*. Each intermediate node that receives the request for the first time appends its identifier to the route accumulated so far in the request, and re-broadcasts the request. When the request arrives to the target, it generates a route reply *(rrep)*. The route reply contains the identifiers of the initiator and the target, the accumulated route obtained from the request, and a digital signature of the target on these elements. The reply is sent back to the initiator on the reverse of the route found in the request. Each intermediate node that receives the reply verifies that its identifier is in the node list carried by the reply, and that the preceding identifier (or that of the initiator if there is no preceding identifier in the node list) and the following identifier (or that of the target if there is no following identifier in the node list) belong to neighboring nodes. Each intermediate node also verifies that the digital signatures in the reply are valid and that they correspond to the following identifiers in the node list and to the target.

If these verifications fail, then the reply is dropped. Otherwise, it is signed by the intermediate node, and passed to the next node on the route (towards the initiator). When the initiator receives the route reply, it verifies if the first identifier in the route carried by the reply belongs to a neighbor. If so, then it verifies all the signatures in the reply. If all these verifications are successful, then the initiator accepts the route.

The proof of the following theorem illustrates how the framework introduced earlier can be used in practice.

Theorem:

endairA is statistically secure if the signature scheme is secure against chosen message attacks.

Sketch of the proof: A routing protocol is statistically secure if it returns a non-plausible route for any *conf* configuration and for any *A* adversary only with negli-

gible probability. More precisely we have to prove that a *rrep* message in the ideal-world model of the protocol is dropped due to its corruption flag only with negligible probability.

Let us suppose that following message is dropped due to its corruption flag in the ideal system, while the real system does not drop it:

$msg = [\text{rrep}, S, D, id, (N_1, N_2, \ldots, N_p), (sig_D, sig_{Np}, \ldots, sig_{N1})]$

In that case we can conclude the following:
- there is no repeating identifier in route $\pi = (S, N_1, N_2, \ldots, N_p, D)$;
- N_1 is a neighboring node of node S;
- every signature is correct;
- S and D are honest nodes;
- every intermediate node (with overwhelming probability) sees the route that was sent by node D (π), since node D signed that route, and every intermediate node checks this signature;
- despite all the above properties, π is a non-plausible route in graph $\underline{G}$, where $\underline{G}$ is the graph of the reduced configuration.

We prove that this can only be possible if adversary A has successfully forged the signature of at least one honest node. We know that there is no neighboring vertices in the graph of the reduced configuration that correspond to neighboring corrupted nodes in the network and in addition each non-corrupted node uses a single and unique non-compromised identifier. It follows that every route, including $(N_1, N_2, \ldots, N_p)$, has a unique meaningful partitioning, which is the following: each non-compromised identifier, as well as each sequence of consecutive compromised identifiers should form a partition.

Let $P_1, P_2, \ldots, P_k$ be the unique meaningful partitioning of the route $(N_1, N_2, \ldots, N_p)$. The fact that this route is non-plausible implies that at least one of the following two statements holds:

1. There exist two partitions $P_j = \{N_i\}$ and $P_{j+1} = \{N_{i+1}\}$ such that both N_i and N_{i+1} are non-compromised identifiers, and the corresponding non-corrupted nodes are not neighbors.
2. There exist three partitions $P_j = \{N_i\}$, $P_{j+1} = \{N_{i+1}, \ldots, N_{i+q}\}$, $P_{j+2} = \{N_{i+q+1}\}$ such that N_i and N_{i+q+1} are non-compromised and $N_{i+1}, \ldots, N_{i+q}$ are compromised identifiers, and the non-corrupted nodes that use N_i and N_{i+q+1} have no common corrupted neighbor.

In Case 1, N_{j+1} does not sign the route reply, since it is non-corrupted and it detects that the identifier that precedes its own identifer in the route does not belong to a neighbor. Hence, the adversary must have forged sig_{Ni+1} in *msg*.

In Case 2, the situation is more complicated. Let us assume that the adversary has not forged the signature of any of the non-corrupted nodes.

N_i must have received

$msg' = [\text{rrep}, S, D, id, (N_1, N_2, \ldots, N_p), (sig_D, sig_{Np}, .., sig_{Ni+1})]$

from a corrupted neighbor, say v^*, since N_{i+1} is compromised, and thus, a non-corrupted node would not send out a message with sig_{Ni+1}. In order to generate *msg'*, node v^* must have received

$msg'' = [\text{rrep}, S, D, id, (N_1, N_2, .., N_p), (sig_D, sig_{Np}, .., sig_{Ni+q+1})]$

because by assumption, the adversary has not forged the signature of N_{i+q+1}, which is non-compromised. Since v^* has no corrupted neighbor, it could have received *msg''* only from a non-corrupted node. However, the only non-corrupted node that would send out *msg''* is N_{i+q+1}. This would mean that v^* is a common corrupted neighbor of N_i and N_{i+q+1}, which contradicts the assumption of Case 2. This means that our original assumption cannot be true, and hence, the adversary must have forged the signature of a non-corrupted node.

Consequently, if a reply message like *msg* can occur in the ideal system with non-negligible probability then the adversary is able to forge the signature of a non-compromised node with non-negligible probability. It contradicts our assumption that the used signature scheme is secure.

4. Summary

In this article, we presented a formal model in which we defined in a precise and rigorous way what we mean by secure ad hoc routing. Using the proposed model, one can prove (or fail to do so) the security of on-demand source routing protocols. We demonstrated the practical usage of the model on a real example, namely, we proved that endairA is secure in our model. In the near future, we will define a similar model to analyze the security of on-demand distance vector routing protocols (e.g. ARAN, S-AODV). Further, we would like to automate the process of proofs by an adequate formal language (e.g. process algebra) and related verification tools.

Acknowledgment

The work presented in this paper has been supported by the Ministry of Education (BÖ2003/70), the Hungarian Scientific Research Fund (T046664) and IKMA.

References

[1] Ács G.: Ad hoc útvonalválasztó protokollok bizonyított biztonsága, TDK dolgozat, 2004. nov. 9.
[2] L. Buttyán, I. Vajda: Towards provable security for ad hoc routing protocols, In Proc. of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington D.C., USA, October 2004.
[3] Y.-C. Hu, A. Perrig: A survey of secure wireless ad hoc routing, IEEE Security and Privacy Magazine, June 2004.
[4] O. Goldreich: The foundations of modern cryptography, Cambridge University Press, 2001.
[5] M. Bellare, R. Canetti, H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols, 1998. In Proc. of the ACM Symp. the Theory of Computing