Weak Authentication: How to Authenticate Unknown Principals without Trusted Parties

Jari Arkko and Pekka Nikander

Ericsson Research NomadicLab,
02420 Jorvas, Finland
{Jari.Arkko,Pekka.Nikander}@nomadiclab.com

Abstract. This paper discusses "weak authentication" techniques to provide cryptographically strong authentication between previously unknown parties without relying on trusted third parties.

1 Introduction

Introducing security mechanisms into environments with little or no existing security, such as the Internet, has proven to be a challenging task. This is mainly due to the practical problems, such as demands for security infrastructure and configuration tasks associated with the used mechanisms. These problems are significant barriers to security deployment, particularly for large networks.

The purpose of this paper is to study "weak authentication" techniques that attempt to bypass some of these problems. In this paper, we use the term "weak authentication" to denote cryptographically strong authentication between previously unknown parties without relying on trusted third parties. We consider this kind of authentication "weak", since the parties involved do not have any "real" information about the identity of their peer.

Thus, at first sight it seems impossible to **authenticate** the identity of the parties in the traditional sense. Surprisingly, however, a number of different techniques can provide some security. None of the techniques are as secure as traditional "strong" identity authentication. It is impossible or at least extremely unlikely for attackers to be able to breach strong authentication techniques. In weak authentication, the probability of a successful attack is also quite small, but not zero.

As discussed earlier by Anderson [2], it is crucial to understand the economic drivers and impacts of various security solutions. Our main research hypothesis is that protocol design should take place with a full understanding of the economic impacts for attacks and defenses, and the residual vulnerability probabilities involved. A secondary hypothesis is that under such analysis, weak authentication techniques offer a better security model for some applications than traditional security methods.

This paper is organized in four parts. First, we need to give an overview of potential applications for weak authentication, and discuss where weak authentication may or may not be applicable. This is done in Sect. 2. Section 3 discusses a

categorization and classification of weak authentication techniques. Section 4 reviews specific techniques for weak authentication and discusses their properties. For instance, a successful attack against a weak authentication method typically requires specific conditions relating to the network topology and the locations of the victims and the attacker over time. Finally, Sect. 5 discusses some tools for understanding the economic aspects and uncertaintities relating to the the residual vulnerabilities in weak authentication methods.

2 Potential Applications

There are many existing and planned applications of weak authentication in some form. Personal area networks are a typical application, such as when we need to connect a mobile phone to a Bluetooth headset. The Secure Shell (SSH) protocol applies weak authentication principles for authenticating servers to users [22]. Users can authenticate their multimedia call signaling using weak authentication in the Session Initiation Protocol (SIP) protocol [18]. The use of weak authentication in providing security for multi-homing, mobility, and other network layer tasks is also being studied [12].

However, weak authentication is only suitable for some applications. In the following, we discuss first some fundamental issues that relate to this suitability.

Some applications require that a real-world identity is presented. Weak authentication is typically more suited for dealing with ephemeral identities. Likewise, weak authentication is less suitable for applications that have real-world impacts. However, a significant number of applications do not have a need for real-world identities, and do not directly affect the real world. Multi-homing, routing, and mobility are typical examples of functions that do not have real-world effects.

The level of protection achieved with weak authentication is also a determining factor. In some special cases – such as for the imprinted duckling described by Stajano and Anderson [19] – a security level comparable to strong authentication is reached. In other cases, it is necessary to compare the remaining vulnerabilities left after weak authentication to those present in the system for other reasons.

In some applications, the cost of strong authentication may exceed the benefits. For instance, Mobile IPv6 route optimization does not justify the introduction of a global Public Key Infrastructure for the sole purpose of authenticating nodes participating in these optimizations. Similarly, some light-weight and low-cost devices may become bulky and impractical if user input devices or tamper-resistance is required as a part of the security solution.

In some other applications that use weak authentication, the cost of attacks to an attacker may exceed the value of the transactions. These applications can, in many cases, be viewed as secure enough.

3 Categories of Techniques

Weak authentication can't be based on pre-shared secrets, as the peers by definition do not have a priori security information about each other. Also, enrollment to an infrastructure is not possible, since that would constitute a trusted third party (or parties), and would require configuration. Traditional Public Key Infrastructure (PKI) or the proposed IETF Authentication, Authorization and Accounting (AAA) infrastructure is therefore out of the question.

One may then ask what is left, and how can any authentication be performed at all? It turns out that there are still several properties that we can use for weaker forms of "authentication" such as *spatial separation*, *temporal separation*, *asymmetric war* on costs, *application semantics*, and *combined* or *transitive* techniques. We discuss each of these separately below.

3.1 Spatial Separation

Spatial separation refers to the ability of a node to ensure that its peer is on a specific communications path. In its simplest form, spatial separation reduces to the requirement of a physical contact, such as the imprinting process proposed for ad hoc wireless networks by Stajano and Anderson [19, 1]. Assuming the underlying communications network is trustworthy to an extent, we can also use challenge-response protocols to ensure that the peer is on a specific path. In a more general setting, a node may view different links and paths as having different degrees of trustworthiness. More than one paths can also be used to further enhance the situation.

3.2 Temporal Separation

By temporal separation we refer to the general ability of the peers to relate communications at time t_1 to some earlier communications at time t_0. That is, there exists some guarantee that the node we talk to at time t_1 is the same one as we talked to at t_0. The guarantees are typically not complete, i.e., they exist only with respect to some assumptions, such as that there have been no man-in-the-middle attackers present at time t_0.

3.3 Asymmetric Costs

As noted by Varian [20] and Anderson [2], economic impacts and interests are often major factors in designing security mechanisms[1]. A troubling observation is that usually it is easier for the attacker to find a single security hole than for the defender to block all holes.

Fortunately, this asymmetric situation can sometimes be reversed and used to the defenders' advantage. This is possible when an individual defender can

[1] Unfortunately, these factors are often calculated from the point of view of a vendor and not the user.

spend a small amount of resources, which then are multiplied on the attacker's side.

The defender can make it harder for the attacker to find the right target. We can also force the attacker to use more resources on an attack even he knows the right target. As an example of the latter, even unauthenticated challenge-response mechanisms work well in some cases [4, 18]. An attacker would have to reveal his location (address) in the network in order to be able to receive the challenge. If the attacker has only a limited number of network locations available for him, he may not be willing to sacrifice them for the sake of being able to attack. More seriously, revealing a network location may also lead to real-world implications for the attacker, such as the network operator or the police taking action against him.

3.4 Application Semantics

Certain applications may offer particular semantics that can be employed to produce weak forms of authentication. For instance, some proposed methods for authorizing Mobile IPv6 route optimization rely on embedding cryptographic information in IP addresses [15, 17].

3.5 Combined and Transitive Techniques

The techniques presented above can be combined. For instance, spatial separation techniques can be combined with temporal separation. An initial physical contact can be used to exchange keys, and leap-of-faith techniques can be used to rely on these later, even over insecure channels.

Web-of-trust models and information from neighbours could potentially be used together with weak authentication techniques to reduce the likelihood of successful attacks. For instance, assume two nodes a and b each have contacted a server c earlier at times t_a and t_b, respectively. Further assume that a and b both can ensure that the node c contacted earlier is still the same c as they are talking to now. This is possible, for instance, if the server gave its public key at the first contact. Then, if a and b trust each other, they verify that they have the same key. Both nodes can then be assured that the server's key has been the same since $min(t_a, t_b)$.

4 Some Concrete Techniques

4.1 Anonymous Encryption

Anonymous encryption is based on temporal separation. In this method, unauthenticated Diffie-Hellman key exchange or some equivalent protocol is run in the beginning of all communication sessions. Here the time period $[t_0...t_1]$ extends only to the length of a single session. This protects the privacy and integrity of the session, assuming there were no active attackers on the communications path at the time the Diffie-Hellman procedure was executed.

Anonymous encryption reduces the likelihood of successful attacks by changing the type of attack required, by requiring attackers to capture more packets, by requiring the attackers to work more in order see communications, and by forcing attacks to be performed before the attacker knows whether a given piece of communication will be useful for it. Nevertheless, the benefits of anonymous encryption for individual users are quite small, except in a statistical sense as we will see later.

4.2 Challenge-Response

Challenge-response techniques can be used to ensure freshness and, under certain assumptions, that the peer is on a specific path towards the given address. It is therefore typically used to ensure spatial separation.

For instance, the Return Routability (RR) method [13, 14, 17] ensures that the peer really is in the place in the network it claims to be in, or at least on a path to it. Such properties may be useful in applications such as Mobile IPv6, which attempts to ensure that modifications made to local routing information do not break existing security properties. While RR provides a low level of protection, it should be noted that so does the current unauthenticated IP routing. Hence modifications to the local routing information can be performed if it can be proven that the node would have been able to get to the traffic anyway. Another example of challenge-response techniques is the use of cookies in TCP SYN packets [4].

This type of weak authentication reduces the risk of successful attacks in a topological sense: out of all nodes in the whole network, only a very small and specific subset can perform an attack.

4.3 Leap-of-Faith

Leap-of-faith is another method based on temporal separation, but it also encompasses aspects from spatial separation and asymmetric cost wars.

Leap-of-faith methods have been successfully employed, e.g., in the SSH protocol [22]. Here the time period [t0...t1] extends from the first contact between the peers to infinity. While in principle similar to the anonymous encryption approach, grouping all sessions under the same protection allows improved security. For instance, the first connection can typically be executed in a safe place. As an example, a Bluetooth earplug and phone can be "paired" in the privacy of the user's home, and the pairing is subsequently secure even in environments that contain potentially malicious parties, such as in a crowded city street [19].

Furthermore, to remain undetected, an attacker must be present at the communications path each time the keys verified through leap-of-faith are being used. For example, if there has been an attacker present when the initial SSH keys have been exchanged, the attacker must be there each time the keys are being used as well. Otherwise the used software would warn the user about a changed key.

The SIP protocol [18] intends to use leap-of-faith methods for multimedia signaling, with the intention that the first call between users establishes self-signed certificates. These certificates would be related to the used identifiers, and would be stored permanently in the phones (guided by user's acceptance).

Role of User Confirmation. Typically, some kind of user confirmation is necessary for leap-of-faith to work in a secure manner. SSH, for instance, authenticates servers by leap-of-faith but the user is prompted to accept the first connection, and warned if the the keys have been changed. (Traditional identification, based on passwords, is used to authenticate the client, which means that the server can proceed unattended.) An imprinting process through a physical contact can also be thought of as a user confirmation.

User confirmation obviously requires at least a rudimentary user interface. The software making use of leap-of-faith also needs the ability to access this user interface, which may not be practical for kernel-level protocols.

Bidirectional leap-of-faith suffers from the problems of getting two user confirmations at the same time. In the case of multimedia phone calls, this problem does not exist as both users must be present for the call to be accepted in any case. In the case of leap-of-faith for general communications this is problematic, since relaxation of the user confirmation rule may create denial-of-service vulnerabilities.

Gehrmann et al have designed a secure user confirmation method to be used in personal area networks [6]. It is required that the devices have a moderately sophisticated user interface. A personal CA device can generate a hash value based on input received over a wireless interface, display this value, and allow the user to enter the same value in the other device for confirmation. We call this type of user confirmation cryptographically strong, as it can not be defeated even by an active attacker. The CA device can also take the role of reducing the need to configure combinations of devices, as the CA produces certificates that can be accepted by other devices in the same network.

State Requirements. In contrast to anonymous encryption, leap-of-faith needs to store state for a long time. Typically, the state consists of an identifier and a key. Where public keys are used, it is possible to store only a hash of the public key and allow the peer to send the public key whose hash we can now verify. Where self-signed certificates are used, even the identifier can be omitted since the hash can cover the whole certificate with the identifier inside it.

When no user confirmation is required, connection attempts (legitimate or fraudulent) from a large number of peers may lead to memory exhaustion. Furthermore, if a real-world identity, such as an IP address, is used, attackers could add just a single fraudulent entry and prevent that particular legitimate peer from establishing a connection later.

A policy is needed for accepting entries for new peers and for purging entries for some peers. Typically, such policy might favor preserving old and frequently

used peer information over recent (and possibly fraudulent) information. Alternatively, if the memory consists of just hashes, lack of memory can be compensated by reducing the length of the hashes by one bit, increasing the risk of collisions but making room for more entries.

Variants of Leap-of-Faith. A technique similar to leap-of-faith can be used with the Host Identity Protocol (HIP) [10, 9, 11]. The verification of the identity happens through the peer offering its identifier, public key, and a signature produced with the private key. What this method provides is a cryptographic proof that the peer really has the given identifier ID. It is therefore not possible for anyone else to claim that their identifier is ID as well. It is, however, possible for anyone to claim that their identifier is ID'. This type of weak authentication is not in general useful to signify what the node in question can be trusted to do. (However, we may already know ID and its authority through some other means.)

4.4 Properties of Identifiers

Recently developed methods can create a strong cryptographic binding between an identifier and its rightful "owner". The cryptographically generated address technique involves the generation of a public-private key pair and a hash of the generated public key [15]. The hash value is used as an identifier. The main application of this is in the area of proving the rightful owners of IPv6 addresses in Mobile IPv6 [17]. The lower host identifier part the IPv6 address is formed from the hash value while the upper part is used for routing and has topological meaning. This method could be applied in other contexts as well, such as in the protection of IPv6 Neighbour Discovery signaling [3].

This method is somewhat stronger than leap-of-faith and HIP-like mechanisms, as the identifier has a dual role for both routing to the right host and as a hash. A verified cryptographically generated address property implies that the peer really is the node who came up with the given IP address first. Since the routing prefix part of the address is not formed through a hash, this applies only to the host identifier part, i.e., the address is right but it hasn't been proven that the routing prefix is really the right one. For this reason, [17] proposes to combine the RR method with cryptographically generated addresses. RR is relatively good for ensuring that the claimed routing prefixes are correct, while cryptographically generated addresses are good for proving the host identifier claim. Together, they can cover the whole claimed IP address.

Weak authentication based on properties of identifiers reduces the likelihood of successful attacks only if the identifiers have a specific meaning for the particular application that uses this authentication. One example of such a specific meaning is the dual role of IP addresses in routing. Where no such specific meaning exists, authentication based on identifier properties reduces to the leap-of-faith method by allowing peers to verify that they are still communicating with the same node as they originally did.

4.5 Opportunistic IPsec

The definition of weak authentication precludes trusted parties. However, a trusted party may have different flavors. Traditionally, when we speak of a trusted party we mean an entity that we have an explicit security association with. Some weak authentication methods use third parties that are "semi-trusted" and do not have such a security association.

A particular approach to weak authentication has been explored in the context of recent "opportunistic security" extensions to the IPsec protocol set [16]. This approach is based on using a public infrastructure – DNS in this case – to supply keys for those parties that wish to employ weak authentication. Communication to other nodes is in the clear. Until DNS becomes secure, this approach is secure as long as one assumes that the path to DNS is free from attackers even if the path to the peer may not be. In practice, this translates roughly to the same security level as anonymous encryption.

As has been pointed out elsewhere, this approach may also lead to unexpected side-effects. In this case, there is an additional mechanism that allows DNS to specify an IPsec gateway to accommodate corporate gateways and other similar devices. An off-path attacker may be able to subvert the DNS mechanisms, e.g., by sending unsolicited responses to the victims, and lure it to use the attacker as the gateway.

5 Modelling Weak Authentication Impacts

5.1 Analysis of Anonymous Encryption

Anonymous encryption defeats passive attackers. However, if a man-in-the-middle is present on the used path, it is likely that this attacker will be able to catch and modify all packets relating to the same session. No memory is retained from one session to another. Therefore, the uncertainty related to the use of anonymous encryption does not change as time goes by. The uncertainty depends only on the probability of a man-in-the-middle attacker being on the used path. This uncertainty is only a bit smaller than the uncertainty of any kind of attackers being present.

However, this analysis considers only the point of view of an individual user. What happens if we consider the whole system view? What if everyone used anonymous encryption? We can analyze this situation by making first a few assumptions:

- Cost of scanning for an interesting transaction consists of equipment needed to monitor traffic on the path, and is 0.1 EUR per transaction.
- Cost of eavesdropping the interesting transaction consists of mainly storage space and analysis costs, and is 1.0 EUR per transaction.
- Cost of performing a man-in-the-middle attack on anonymous encryption consists of equipment needed to intercept traffic and run Diffie-Hellman in both directions. It is 10 EUR per transaction.

– There is one interesting transaction per one million transactions, or one interesting person in a population of a million.

We can now analyze the economic impacts for the three following cases:

1. There isn't anyone who uses anonymous encryption. In this case the attacker's costs are 0.1 EUR $*1,000,000 + 1$ EUR $= 100,001$ EUR.
2. Only the interesting person uses anonymous encryption. In this case the attacker's costs are 0.1 EUR $*1,000,000 + 10$ EUR $+1$ EUR $= 100,011$ EUR.
3. Everyone uses anonymous. In this case the attacker is forced to use an expensive scanning method (MitM) against everyone, raising the costs considerably: 10 EUR $*1,000,000 + 1$ EUR $= 10,000,001$ EUR.

The attacker considers these numbers as follows. The attacker must have sufficient resources to commit to the attack, and the potential benefit from the attack must be great enough to satisfy the expenditure.

For instance, if implemented widely for all HTTP connections, anonymous encryption would prevent widespread snooping, unless all communications would be routed through a device capable of a Diffie-Hellman man-in-the-middle attack for all sessions. Since the cost of such a device would be prohibitive, and the routing and delay aspects would probably give away this activity, the result is that all communications would be statistically more secure than today.

The cost of attacks and defenses must also be considered by the legitimate users. We need to consider the motives for the users to to implement defenses. Our defense is not directly useful for this particular user. Some users might not want to use the defense, particularly if it is costly. Such a situation is called the "tragedy of the commons" [8]. Fortunately, in many cases this situation can be avoided. For instance, a protocol may be designed in a standards origanization, and it is often possible to mandate a specific defense mechanism.

In conclusion, while techniques like anonymous encryption are not useful for a single individual, they may be useful for the community as a whole. This conclusion is, however, dependent on the assumption that the attacker is interested on a particular person. If the attacker is satisfied just with finding anyone to attack, the conclusion no longer applies. We should also note that the attackers and legitimate users often do not have equal resources. For instance, hand-held low-power devices have limited CPU resources, but an attacker that controls a number of virus-infected personal computers may have a large amount of CPU resources.

5.2 Analysis of Challenge-Response

Challenge-response mechanisms use spatial separation and ability of only a limited number of attackers to see the challenge. Assuming each path has some small probability (say 0.1) of having an attacker, challenge-response mechanisms leave this level of uncertainty with its authentication result. This uncertainty

can be compared to the probability of *some* path in the Internet having an attacker, which is 1. A node that does not use challenge-response and accepts commands from other nodes directly has the uncertainty of 1, while a node that uses challenge-response has an uncertainty of 0.1.

Economic analysis of challenge-response mechanisms is also possible. Consider the application of Mobile IPv6, and the use of the RR method. Assume that the attacker is interested in causing as large distributed denial-of-service attack as possible. Where would this attack be easiest to perform? Obviously at the core of the network, as the attacker could then be in a place where the number of nodes using path with support for RR is the largest. We can say that the value of the location is $nodes * capacity$, where *nodes* is the number of nodes that view this location as the path towards some useful attack address, and *capacity* is the bandwidth available at the location for the attacker.

5.3 Leap-of-Faith Analysis

Leap-of-faith uses temporal and spatial separation, and to a smaller extent also asymmetric cost wars. A simple model for the uncertainty related to leap-of-faith is that each path where it is used has a probability (say even as high as 0.9) of having a man-in-the-middle attacker. Let us further assume that the total number of different attackers in the network is 2. Then on the first use the uncertainty of the method is 0.9, on the second use $0.9 * 1/2 = 0.45$. In general, on the kth use the the uncertainty is $0.9^k * (1/2)^k$. Here the first component represents the need for the attacker to be present on all communication attempts in order to escape detection of the attack. The second component represents the need for the attacker to be the same on all uses.

6 Previous Work

Much literature exists in all particular areas of weak security discussed above, such as how MIPv6 works without trusted parties. Many weak authentication mechanisms have already been defined and deployed, such as those used by TCP, SSH, or SIP.

However, we are not aware of any work that tries to classify and analyze different types of weak authentication techniques in a general setting. Stajano and Anderson have discussed various techniques that are suitable for one application area that appears suitable for weak authentication techniques, Ad Hoc wireless networks [19]. Gehrmann et al have extended this work towards different kinds of devices in a personal area network [6].

Anderson has discussed the economic impacts related to security mechanisms [2], but has not taken the step we are taking here to design protocols based on these impacts.

Josang [7] Yahalom [21], Beth [5] and others have studied formal models for understanding trust and uncertainty.

7 Conclusions

We have reviewed weak authentication techniques and discussed their properties. Our main conclusions are:

- Sometimes imperfect security can be good enough for the task at hand, or even provide excellent security.
- The cost-benefit curve is not linear. Some basic techniques can provide a significant advantage with little or no cost. An example of this is the TCP SYN cookies method [4]. These techniques can typically used all the time, and stronger security can be used where necessary or possible.
- Results of uncertainty, probability, and economic impact analysis are often surprising.
- Protocols should be designed based on the understanding of the issues relating to uncertainty and economic impacts, in a given application setting.
- Some techniques may lead to the "tragedy of the commons" situation. This can be avoided if standards organizations can mandate certain techniques as a part of a protocol specification.

Further work is needed at least in relating weak authentication and its models to the notion of trust. We also need to create formal models to describe trust, uncertainty, economic impacts, and a view to the whole system. Individual weak authentication methods should be studied further and new ones need to be developed for new applications. The role of multiple contacts, user confirmation, and transitive and combined methods needs more work. The continuum between no user confirmation and cryptographically strong user confirmation is also of particular interest.

References

1. Frank Stajano Anderson. The resurrecting duckling: What next? In *8th International Workshop on Security Protocols*, Cambridge, UK, 2000.
2. Ross Anderson. Why information security is hard - an economic perspective. In *Proceedings of the 17th Annual Computer Security Applications Conference*, December 2001.
3. Jari Arkko, Tuomas Aura, James Kempf, Vesa-Matti Mantyla, Pekka Nikander, and Michael Roe. Securing IPv6 Neighbour Discovery. unpublished manuscript, submitted for publication, 2002.
4. D. J. Bernstein. Syn flooding. http://cr.yp.to/syncookies/idea, 1996.
5. Thomas Beth, Malte Borcherding, and Birgit Klein. Valuation of trust in open networks. In *Third European Symposium on Research in Computer Security (ESORICS 94)*, 1994.
6. Christian Gehrmann, Kaisa Nyberg, and Chris J. Mitchell. The personal CA - PKI for a Personal Area Network. IST Mobile Summit 2002, 2002.
7. Audun Josang. Trust-based decision making for electronic transactions. In L. Yngstrom and T. Svensson, editors, *Proceedings of the Fourth Nordic Workshop on Secure IT Systems (NORDSEC'99)*, Stockholm University Report 99-005, 1999.

8. W. F. Lloyd. Two lectures on the checks to population. Oxford University Press, 1833.
9. Robert Moskowitz. Host Identity Payload and Protocol. Internet Draft draft-moskowitz-hip-05.txt (Work In Progress), IETF, November 2001.
10. Robert Moskowitz. Host identity payload architecture. Internet Draft (expired) http://klovia.htt-consult.com/draft-moskowitz-hip-arch-02.txt (Work In Progress), IETF, February 2001.
11. Robert Moskowitz. Host Identity Protocol implementation. Internet Draft (expired) http://klovia.htt-consult.com/draft-moskowitz-hip-impl-01.txt (Work In Progress), IETF, February 2001.
12. Pekka Nikander. Denial-of-service, address ownership, and early authentication in the IPv6 world, April 2001.
13. Pekka Nikander and Charlie Perkins. Binding authentication key establishment protocol for Mobile IPv6. Internet Draft draft-perkins-bake-01.txt (Work In Progress), IETF, July 2001.
14. Erik Nordmark. Securing MIPv6 BUs using return routability (BU3WAY). Internet Draft draft-nordmark-mobileip-bu3way-00.txt (Work In Progress), IETF, November 2001.
15. Greg O'Shea and Michael Roe. Child-proof authentication for MIPv6 (CAM). *Computer Communications Review*, April 2001.
16. M. Richardson, D. Redelmeier, and H. Spencer. A method for doing opportunistic encryption with IKE, October 2001.
17. Michael Roe, Greg O'Shea, Tuomas Aura, and Jari Arkko. Authentication of Mobile IPv6 binding updates and acknowledgments. Internet Draft draft-roe-mobileip-updateauth-02.txt (Work In Progress), IETF, February 2002.
18. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. Internet Draft draft-ietf-sip-rfc2543bis-09.txt (Work In Progress), IETF, February 2002.
19. Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *7th International Workshop on Security Protocols*, Cambridge, UK, 1999.
20. Hal R. Varian. Managing online security risks. *The New York Times*, June 2000.
21. R. Yahalom, B. Klein, and Th. Beth. Trust relationships in secure systems - a distributed authentication perspective. In *Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy*, pages 150–164, 1993.
22. T. Ylonen, T. Kivinen, M. Saarinen, T. Rinne, and S. Lehtinen. SSH protocol architecture. Internet Draft draft-ietf-secsh-architecture-12.txt (Work In Progress), IETF, January 2002.