Who do you trust?
Trust Hierarchies Over Insecure Communications Channels

Adam Fields Matthew Ringel

4 May 1995

Abstract

Current developments in trust systems have proven to be ineffective on a large scale by the simple fact that they have failed to gain widespread use. Of these developments, three models have come to the forefront and have achieved some measure of usage— "The Web of Trust", "The Strict Hierarchy Model", and "The Escrow Encryption Standard". This paper proposes to examine these three methods in terms of usage in the growing worldwide network and to explain why they are inadequate. Additionally, we will propose a new model that possesses the scalability necessary for a large user base and the scope of worldwide network interconnections, as well as an implicit understanding of the methods used in "the real world" for establishing trust. We will examine the technological and sociological ramifications of implementing our system, as well as any problems that remain to be solved, through a variety of examples of how the system could work in practice.

Part I

Introduction

The determination of trust between two entities who desire to communicate securely over an insecure channel, as well as the quantification and authentication of that trust, is a difficult problem that gets worse every time a new user or system joins the network community. Several methods have been proposed and employed to attempt to dictate a viable trust model, with varying degrees of success. However, none has yet approached an adequate mix of security, scalability, and end-user ease of use that would make it truly usable, effective, and accepted into general use on a world-wide scale.

Of the three primary existing systems, only two are in common use — the "Web of Trust" (WoT), as implemented by PGP, and the "Strict Hierarchy Model" (SHM), as implemented by many military, governmental, and corporate organizations. The third, the "Escrowed Encryption Standard" (EES) has as yet failed to gain a foothold in any practical large-scale application, for good reasons, as will be explained later.

One of the fundamental problems in crafting a trust system is that the very definition of the word "trust" is vague in its application to electronic security. The best way to define trust is not by an isolated definition of the word, but in its applications to pre-existing sociological constructs. Eisenstadt and Roniger mention in
[Eisenstadt & Roniger, PATRONS, p. 21-22] that:

> "The structural-functional school or approach addressed itself squarely to the problem of how the dimension of solidarity (trust), meaning, and – to a smaller degree – power is institutionalised in the construction (or 'production') of social order. These dimensions of social life were defined as needs which every social system (and in a different way also personalities and cultures) has to cope with, or as prerequisites of the inherent working of such systems."

This can be extrapolated to include the electronic community. When communicating with unknown others in "the real world", there are certain clues that one can use to determine whether or not to trust another person. In a non-electronic communications medium, the scope of interpersonal connections is often small, and references become much more important. Additionally, communicating face-to-face (or voice-to-voice) allows the interested party to use semantic cues to determine trust (appearance, tone of voice, etc...). In an electronic medium, these are, for the most part, absent, and the user must rely on an entirely different set of criteria for trust establishment.

We believe that an effective model can be crafted by combining the organizational concreteness of the SHM as well as the low-level granularity and redundancy of the WoT, while providing a well-defined set of criteria whereby trust may be established. It is our belief that by creating a "Web of Hierarchies" (WoH), a system that is both scalable and yet still manageable on a peer-to-peer level can be implemented.

Part II

Inadequacies in the current systems

2.1 Design requirements for a usable, large-scale trust model

In order for a trust model to be implementable on a large scale, it must have the following properties:

Accessibility It must be easily accessible to every end-user in the system. As seen by the use, or lack thereof, of PGP, if a system is not easily usable, people will simply fail to make the effort.

Accuracy A user's "trusted status" in the system should be up-to-date based on past performance, via a combination of external agencies. In addition, the gradations of trust should be fine enough for accurate comparison, but coarse enough to provide meaningful differences between the trust levels. As a corollary to this, the trust levels should provide some indication of the physical presence/authenticity of other users.

Speed A user should be able to assess the trusted status of another user in a timely fashion. This is related to accessibility.

De-centralization The trust hierarchy should be de-centralized, in order to reduce strain on the network and increase the chances of surviving small-scale catastrophic failures.

Easily enterable Establishment of initial trust within the model should be easy. Again, this is related to accessibility.

Scalability The system should remain effective with any amount of participants that wish to communicate with each other, be it ten, a thousand, or possibly several millions.

Choice The system should exist only to provide information to the end-user about who to trust, and should not deny the user any ability to communicate with whom they choose.

It should be noted that some of these points conflict, and a truly effective model must balance them all. For example, emphasis on de-centralization or accuracy will tend to decrease the response time of the system, possibly to the extent of making unusably slow. In addition, as evidenced (in a completely unrelated incident) by the Apple Macintosh Operating System, making a system easy to use for new users often results in the lack of additional features for experienced users. In short, the system should have more features available for those who want them, but should be usable on a low level by everyone.

2.2 Failures in the Web of Trust model

"You don't know about me? Just ask my friends Boris and Natasha."

Simson L. Garfinkel provides an excellent introduction to the WoT concept—

> "Most of us start life knowing just a few people–the members of our immediate family, perhaps a babysitter, and some playmates. As we grow older, we meet more people. Some of them are trustworthy: we can count on them not to steal our possessions, not to hurt us, and to help us up when we fall down. Other people aren't too trustworthy (and the less said about these people, the better).
>
> How do we know whether to trust the new people that we meet? Most children trust everyone. But as they grow older, they become suspicious. After just a few years, when kids meet new people, kids look to their parents and their friends–people they already trust–to figure out whether they should trust the newcomers."[Garfinkel, PGP]

The Web of Trust model is viable only when used by small, self-contained groups of people, for the following reasons:

- It is dependent on the existence of an easily accessible path in the WoT between the two parties.
- Each user must, on an individual level, keep track of who they trust and who they don't at all times.
- Trust is usually established without possibility of a disinterested third party.
- It is usually difficult to find more than one independent agent who will verify the trust status of a particular person.

The WoT, as evidenced by its implementation under PGP, fails to meet several of the above requirements. It is difficult and daunting to use for new users because the entire responsibility for establishing and maintaining trust is placed in the hands of the end-user, who is often not ready or able to handle such responsibility. Additionally, there is no easy way to update trust within the system. Once a key is signed by a user, the only way to remove that signature is to revoke either the signing key or the signed key. This could lead to keys that have been validated, but are no longer valid. An additional problem with the WoT model is that a user needs to store for him or herself a separate key for each party with whom he or she wishes to communicate. This is clearly unscalable to the scope of a worldwide network, and is more popularly known as the "n-by-n representation problem". In a similar vein, as a result of the process used to verify public keys with digital signatures, multiple normally uninvolved parties are required in order to fully verify a public key. For even a single two-way conversation, this can lead to a tangled web of verification requests, as seen in Figure 1.
While the WoT does have its drawbacks, it does have certain fundamental characteristics in its favor, which we have kept in mind when designing our own system. It is

non-centralized, allowing the required information to be stored in pieces in many different places and preventing it from being easily compromised, and it is easy for a new user to establish basic trust and gain more as time goes on, although this does not insure the ability to communicate with everyone.

2.3 Failures in the Strict Hierarchy model

"You don't know about me? Well, then you can't trust me until my government annexes yours."

> "The Orange Book talks about 'trusted' systems rather than 'secure' systems. The words aren't really synonymous. No system is completely secure. Any system can be penetrated—given enough tools and time. But systems can be trusted, some more than others, to do what we want them to do, and what we expect them to continue to do over time.
>
> The central concept of the Orange Book is that it's possible to measure this trust—to build, evaluate, and certify a system that conforms to a specific set of security criteria, and that therefore merits a certain overall security rating."[Russell & Gangemi, CSB, p. 105-106]

The Strict Hierarchy Model, as outlined in [U.S. DoD, OB], and as implemented by various government, military, and corporate institutions, is not really viable at the granularity of an individual user in a world-wide network environment. Since it relies on pre-defined multilevel standards of who gets trusted when and by whom (see figure 2), the individual user is left in the dark, with no easily established place in the system. In some Orwellian futures, the world may be ordered this way, but for now it is not. Additionally, the trusted status of others in relation to a particular user has very little to do with the users themselves. This model centers trust relations in the interests of the company (or government) rather than the individual user, with trust qualifications dictated from further up the line. Although it scales extremely well to multi-user networks, it is not viable for a world-wide personal network because of this requirement for already established trust lines. Additionally, the model fails on the following issues:

- Users in the system are not permitted to make their own choices regarding who to trust – all such decisions are dictated by the model itself.
- There is no de-centralization. This is not necessarily a problem, but there is the possibility of a single point of failure which could affect the entire system. In a military setting, this is not much of a concern, but in a setting where there is communication between autonomous parties, where there is no hierarchy imposed externally, it could be disastrous.
- While it is utterly scalable, this scalability is completely dependent on the external dictation of trust status which is absent from everyday interpersonal relations.
- "The Orange Book focuses on only one aspect of security—secrecy—while paying little attention to the principles of accuracy, availability, and authenticity."[Russell & Gangemi, CSB, p. 112]

2.4 Failures in the Escrowed Encryption Standard model

"You don't know about me? Oh, just ask the government."

The Escrowed Encryption Standard operates as a hardware-only encryption scheme between two parties who both possess EES-capable devices. While this restriction would normally not be a major issue, EES also has, built into it, the backdoor that some third party, in most cases the government, can gain access to the session keys used in any given transmission.

> "By far the most controversial aspect of the EES system, however, is *key escrow*. As part of the crypto-synchronization process, EES devices generate and exchange a "Law Enforcement Access Field" (LEAF). This field contains a copy of the current session key and is intended to enable a government eavesdropper to recover the cleartext. The LEAF copy of the session key is encrypted with a device-unique key called the "unit key", assigned at the time the EES device is manufactured. Copies of the unit keys for all EES devices are to be held in "escrow" jointly by two federal agencies that will be charged with releasing the keys to law enforcement under certain conditions."[Blaze, EES]

The EES is very different from the preceding two examples. Not only is it impractical for the average end-user, it is fraught with requirements that most people would cringe at if asked to use them in "real world" communications (see figure 3). It is too politically biased to be of any use on a world-wide scale. For example, if someone in the United States wants to communicate with someone in Europe, using an EES device, the person in Europe must have a U.S. Government-approved EES device. Since that other person is not a U.S. citizen, they do not have the protections normally associated with the escrowing of keys by the U.S. Government, and as such, their keys would be readily accessible by any government agent that cared to look.
Anyone who uses the standard must adhere to a specific encryption algorithm[1], must be communicating with someone who also has an EES device, and must absolutely trust the governmental agency to keep their keys secret. With all of the publicity that computer crime has been getting lately, this is not a good bet.
To summarize, the EES is not viable on a world-wide scale because:

- It is not usable by anyone outside of the United States, for the reasons mentioned above.
- The system provides no information about who any given user is communicating with, other than the fact that there exists a secure communication. As such, it is not viable for establishing peer-to-peer trust, but only for secure communications among already trusted parties.
- This system is dependent on a specific algorithm. This is a danger in itself, but when coupled with the secrecy of the algorithm and lack of widespread scrutiny, raises disturbing questions about the stability of the system.

[1] Which, incidentally, has been shown to be insecure in some ways.[Blaze, EES]

- Much like the SHM, the user is forced to arbitrarily trust an agency (or agencies), in this case the escrow agents, that have access to the means to decrypt transmissions at will.
- The only implementation that has been proposed for widespread use — *Clipper* — has been shown to be infeasible, even for its intended purpose.[Blaze, EES]

Part III

The Web of Hierarchies model

3.1 Description of the model

The WoH is modeled on the structure of the DNS[Postel, DNS]. By following this built-in hierarchy, the trustmaster web is conceptually obvious, as well as efficient. Many of the problems associated with the WoT are eliminated by adding a trusted chain to a central source (for the United States, the NIC) for key verification. The Web of Hierarchies (WoH) model combines the structure and speed of the SHM, with the flexibility and choice of the WoT. The following example illustrates the bulk of the procedure: (Please see figure 4)

1. Bob, a user at a random company, has an important letter, regarding the defection of several employees from Alice's company, Alice included, to Bob's company. He'd like to send it to Alice, a user at another random company. He has no information about Alice other than her email address and her real name. Before he sends the letter, he'd like to verify that alice@dept.alice.com is actually Alice Lionheart, the actual intended recipient of the letter, and he'd also like to get Alice's verified public key, so he can ensure that only she can read the letter. Because several skilled members of Bob's company don't like anyone from Alice's company or, for that matter, Bob, he cannot get Alice's public key from Alice herself, because it would just be replaced by one of theirs, and Alice, Bob, and everyone mentioned in the letter would be summarily executed.

2. Bob has a solution, however. He sends a request to trustmaster@dept.bob.com via a previously arranged private-key encrypted connection, asking for information about alice@dept.alice.com.

3. trustmaster@dept.bob.com has never heard of alice@dept.alice.com, so it sends a request for the same information to trustmaster@bob.com via a private-key encrypted connection.

4. trustmaster@bob.com has never heard of that address either, but it does have the public-key of trustmaster@alice.com, because people from the two companies are constantly exchanging secure hate-mail. It sends a request for information about alice@dept.alice.com to trustmaster@alice.com.

5. trustmaster@alice.com has never heard of alice@dept.alice.com, so it forwards the message, via a private-key encrypted connection, to trustmaster@dept.alice.com.

6. This machine has heard of alice@dept.alice.com, and has information about her. It forwards all of the information in its database regarding that user account back up the

chain and back to Bob. This information includes Alice's public key, and a report from trustmaster@dept.alice.com stating that alice@dept.alice.com is the userid of Alice Lionheart, and has been verified in person with a valid driver's license and advanced class ham radio license, and that she has been the owner of this account since April 7th, 1962.

7. Bob, satisfied with the report from the trustmaster, decides to take the risk and send his encrypted message to Alice, and the defectors all live happily ever after, while Alice's former company is reduced to a small dust cloud the size of Alice's left middle toe.

This example illustrates the salient points of the system. Although it seems like such a system would be slow during actual implementation, certain paths can be eliminated on repeat trips through the system. To speed things up, trustmaster@bob.com can save the public key for any trustmasters contacted, for a preset time, so the NIC will not have to be contacted for repeated transactions with the same host. Also, Bob can store Alice's public key for future transactions, bypassing the web entirely. Bob will only need to go through the system again if:

1. he wants to talk to someone else.
2. Alice's key becomes suspect for some reason and he wants to re-verify it.
3. he's running out of disk space and can't afford to store all of the keys he's accumulated.

These last two points illustrate the main differences, in practice, between the WoH and the WoT. The storage method of the WoH generalizes to that of the WoT when users keep their own collection of keys, but this is not a necessary condition for using the system. If a user decides he wants to speed things up, he can do so by storing keys, but if he decides that he doesn't want to spend the disk space, he can always, with one message, re-retrieve any key he's gotten in the past. Unlike the WoT, the WoH incorporates a variation on the concept of an arbitrator to complete the protocol.

> "An **arbitrator** is a disinterested third party trusted to complete a protocol... Disinterested means that the arbitrator has no particular reason to complete the protocol and no particular allegiance to any of the people involved in the protocol. Trusted means that all people involved in the protocol accept that what is said is true, what is done is correct, and that his or her part of the protocol will be complete... In the real world, lawyers are often used as arbitrators."[Schneier, APPLIED, p. 22]

In the WoH, the trustmasters can be viewed as lawyers acting in the interests of their clients, Bob and Alice, and the NIC acts as the arbitrating judge, negotiating contacts between the two lawyers. Like in the real world, if the lawyers (trustmasters) can reach a settlement agreeable to both parties, the intervention of a judge (the NIC) is not required, and if Bob and Alice can work it out themselves, they have no need to involve the lawyers (trustmasters). The jury is still out.

3.2 Requirements fulfillment assessment

As noted earlier, these requirements are, in some cases, contradictory. Unlike the other models, the WoH takes each requirement into consideration, and succeeds in balancing them with a minimum of compromise.

Accessibility Every user has an associated trustmaster that is at their level of the DNS. Every transaction beyond that initial request is transparent to the user. This initial transaction could also be automated for novices such that it would automatically take place if the user didn't have the desired public key already stored.

Accuracy Since complaints can be registered with the trustmaster via email and physical authentication information is stored by the trustmaster, the system will be as up-to-date as its administrators choose to keep it. It will be to their advantage, and the advantage of their users, to keep it as new as possible, in order to encourage others to do the same.

Speed The system will perform slowly at first, with a high load on the NIC key database. As trustmasters start accumulating a public key database, perhaps with some sort of "least recently used" cache, the load on the NIC will drop, and the system will settle into normal operation. Email communications up the hierarchy will be very fast, with the only real bottleneck being the communications between the trustmasters at the highest level. Information can then be obtained in essentially the time it takes to send a message to the remote system and receive a response.

De-centralization While all trustmasters must initially access the NIC database, as requests for information come in, the load will drop, and trustmasters will be able to communicate efficiently among themselves. Since public keys are stored remotely, at each user's site, the compromising of one site only causes those public keys to be lost. Security could be increased by permitting access to trustmaster files only by the trustmaster, and no other account.

Easily enterable Each user will be given a private key to be used with the trustmaster at his level and a space to store his public key for trustmaster access. This should be set up in person, preferably when the account is created, but could be done by certified mail of some sort. If it is done in person, trustmaster verification of identity could occur at the same time.

Scalability Because the system is modeled on the DNS, the scalability is built-in. If export restrictions on cryptography are ever lifted, some counterpart to the NIC for foreign parties could serve the same purpose, making the system accessible on a world-wide scale. On a national level, as new systems are created, they are simply added to the NIC database, and gain instant entrance to the web.

Choice Regardless of whether communications with the trustmaster are automated or not, the choice to trust the data supplied always lies with the user. If a public key is available and verifiable, it is passed on to the user, who can then decide whether or not to use it.

3.3 Implementation Considerations

The natural choice for implementation of this system is PEM (Privacy Enhanced Mail). The only change in the PEM protocol that we would advise is the replacement of X.500 naming specifications with the much-simpler structure of the DNS[Schneier, EMAIL, p. 284]. The reason for this is because X.500 doesn't scale down well, in that its naming structure is too lengthy and awkward for use with anything smaller than a planetary scale (n.b.: the scalability requirement includes scaling down as well as scaling up). Additionally, the DNS hierarchy is already in place and being used on a world-wide scale. Attempting to replace this with other designations is becoming more and more infeasible as the Internet grows. The time has long past for another "Flag Day".[Raymond & Steele, HACKER] As noted elsewhere in this paper, provisions for trustmaster management are left up to the administrators of individual systems, although it is reasonable to believe that the actual trustmaster address will be primarily administered by an electronic agent, except in the cases where human intervention is required (e.g.: physical authentication of users; negotiation of key exchange with the NIC, with individual users, and between trustmasters in the same domain; etc...)

3.4 Potential Attacks & Weaknesses

The WoH protocol is subject to all standard denial-of-service attacks, unless care is taken to physically safeguard the lines of communication. As of now, a redundancy scheme has not been worked out, but it would be easy to add because of the inherent de-centralization, making the system somewhat more resistant to denial-of-service attacks. The man-in-the-middle attack is circumvented by the creation of a cryptographically secure path throughout the entirety of the web/hierarchy. Replay attacks are still a danger, but would only result in system load as repeated messages are sent containing requests for information. The communications path is only as secure as the cryptographic schemes used, however, so care must be taken to use only those algorithms that can not be easily defeated. The only real danger that we can find lies in the possibility that a trustmaster site may be compromised. If this were to happen, all information coming out of that site would be suspect. To decrease this danger, trustmaster files could be stored using a cryptographic file system, such as the one outlined in [Blaze, CFS].

Part IV

Conclusions

The need for a scalable, effective, trust establishment protocol on the Internet is clear. That existing models are ineffective on this scale, for various reasons, is also clear. The Web of Hierarchies model provides a solution that fulfills and balances the stated design requirements, is politically acceptable (with the notable exception of export restrictions on cryptography) and, above all, could be implemented relatively non-invasively, given the current state of the Internet. This is important because, even now, the Internet has grown to a size such that sudden massive administrative changes are not just impractical, but almost impossible. As such, any system which is to be implemented on a world-wide scale must be as transparent as possible.

Bibliography

[Garfinkel, PGP] Simson Garfinkel
PGP: Pretty Good Privacy
O'Reilly & Associates, Inc.
USA, 1995.

[Blaze, EES] Matt Blaze
Protocol Failure in the Escrowed Encryption Standard
AT&T Bell Laboratories
August 20, 1994.

[Schneier, EMAIL] Bruce Schneier
E-MAIL Security: How to keep your electronic messages private
John Wiley & Sons, Inc.
USA, 1995.

[Russell & Gangemi, CSB] Deborah Russell & G.T. Gangemi Sr.
Computer Security Basics
O'Reilly & Associates, Inc.
USA, 1991.

[Schneier, APPLIED] Bruce Schneier
Applied Cryptography John Wiley & Sons, Inc. USA, 1994.

Great Britain, 1963.

[Eisenstadt & Roniger, PATRONS] S.N. Eisenstadt & L. Roniger
Patrons, Clients and Friends
Cambridge University Press.
Great Britain, 1984.

[U.S. DoD, OB] U.S. Department of Defense
Department of Defense Trusted Computer System Evaluation Criteria
U.S. Government Printing Office. Washington D.C., 1985.

[Postel, DNS] Jon Postel

Domain Name System Implementation Schedule – Revised
Internet RFC 921.
October, 1984.

[Blaze, CFS] Matt Blaze
A Cryptographic File System for Unix
AT&T Bell Laboratories.
November, 1993.

[Raymond & Steele, HACKER] Eric Raymond & Guy Steele
The Hacker's Dictionary
MIT Press.
Cambridge, 1992.