The PGP Trust Model

Alfarez Abdul-Rahman

{F.AbdulRahman@cs.ucl.ac.uk}
Department of ComputerScience, University College London
Gower Street, London WC1E 6BT, United Kingdom

In relation to the cryptographic and legal issue of PGP, its trust model is its lesser scrutinised aspect. How PGP automatically derives "trustworthiness" is important, as wrongful underlying assumptions can undermine its effectiveness. In this paper, we shall attempt to explore the fundamentals of the model, and suggest some improvements.

1. Introduction

Pretty Good Privacy [3], or PGP, is a milestone in the history of cryptography, because for the first time it makes cryptography accessible to the wide mass of privacy hungry on-line public. PGP was created primarily for encrypting e-mail messages using public or conventional key cryptography. The latter are used mainly to encrypt local files. With public key cryptography, PGP first generates a random session key and encrypts the plaintext with this key. The session key along with the ciphertext are then encrypted using the recipient's public key and then forwarded to the recipient. Other features include generating message digests, generating digital signatures, management of personal 'key rings' and distributable public key certificates. It is also designed to work off-line to facilitate e-mail and file encryption, rather than on-line transactions.

One of the major problems of PGP has to do with key distribution and management. The lack of fixed or formal certification paths means that the uncertain authenticity of any PGP key certificate becomes a rather significant matter. This, along with other issues which are more implicit in the design and structure of PGP must be scrutinised and made clear, so that the layman may be able to use this powerful tool effectively, and more importantly, without any incorrect assumptions.

This document analyses aspects of PGP[1] that are related to its trust model. The mechanism for ascertaining trust in PGP is presented, and then underlying assumptions about the protocol and its implications are discussed. It is assumed that the reader is equipped with the general understanding of the use of public-key certificates and protocols.

2. Trust Architecture

Following its intention as a 'cryptographic tool for the masses', PGP breaks the traditional hierarchical trust architecture and adopts the "web of trust" approach [2]. There are no central authority which everybody trusts, but instead, individuals sign each other's keys and progressively build a web of individual public keys interconnected by links formed by this signatures.

For the purpose of illustration, lets assume that Carol, a researcher, requires some data from a source named Bob whom Carol have never met before.

Alice, a colleague of Carol's, signs Bob's public-key certificate which she knows is authentic. Bob then forwards his signed certificate to Carol who wishes to communicate with Bob privately. Carol, who knows and trusts Alice as an *introducer*, finds out, after verification, that Alice is among Bob's

[1]PGP version 2.6.3i was the latest version at time writing, which was also the version used in the preparation of this paper.

certificate signer (Bob could have more than one signature on his certificate to make it more widely acceptable). Therefore, Carol can be confident that Bob's public key is authentic. However, had Carol not known or trusted any of Bob's signers, including Alice, she would have been skeptical about the authenticity of Bob's public-key. Bob would have to find another introducer whom Carol trusts to sign Bob's public-key certificate. This is illustrated in Figure 1, where the dotted arrow indicates "Carol trusts Alice as introducer" and the solid arrow indicates "Alice trusts Bob's public key validity". These distinctions are discussed later in Section 3.

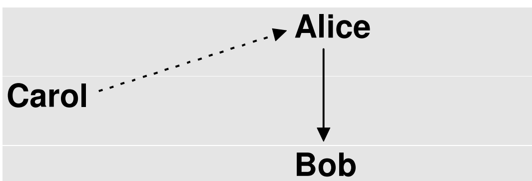

Figure 1 Alice introduces Bob to Carol by signing Bob's public-key certificate before Carol receives it.

A by-product of this approach is the emergence of communities of trust webs, mirroring the tight inter-relationships within social groups of various categories (eg. kinship or occupational groups), and the looser inter-community relationships. Some of the individuals within a community may have trusted friends in another community, so this sort of trust-relations could form a bridge between communities.

As an example, Figure 2 may represent two groups of people: Carol, Alice, David and Greg are researchers who work together. Bob Eric and Fred represent commercial organisations which provide field data, and it happens that Alice and Greg uses Bob as a supplier for their research data.

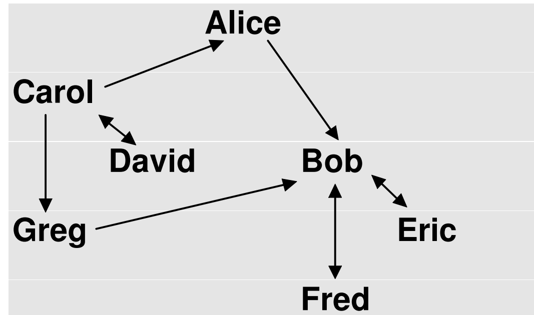

Figure 2 "Communities of trust".

3. Trust notion in PGP

Public key certificates are central to PGP. Each certificate contains the key owner's user ID, commonly represented by the owner's e-mail address in the form "name <userid@domain>", the public key itself, a key ID and date and time of creation. All of this may be digitally signed by any number of "introducers". In Example 1, Carol trusts Alice to introduce Bob, so, Carol regards Alice to be an introducer for her. Carol may have as many introducers as she likes. It is worth noting here that each entity in the PGP web of trust is primarily identified by her public-key, and not the ID or "name" attached in the certificate. This is because names can be arbitrarily assigned to a public key, and it not necessarily be the entity's "real" name. Hence it is a widely accepted practice within the PGP community to refer to a public key by its *key-ID* rather than its owner-ID. Each key-ID is globally unique and included along with the certificate. The only problem is that key-IDs are harder to remember than owner-IDs. For example, <alice@research.com> is easier to remember than its key-ID, which typically looks like *1336F7*. There are two areas where trust is explicit in PGP:

1. Trustworthiness of public-key certificate.

2. Trustworthiness of an introducer.

3.1 Trustworthiness of public-key certificate

This says whether a PGP public-key *certificate*[2] is reliable or not. In other words, whether we can be confident with the binding between the ID and the public-key itself, both contained in the certificate. There are various degrees of confidence attached to a certificate's validity. These are categorised roughly in PGP as follows:

- *undefined*: we cannot say whether this public key is valid[3] or not.
- *marginal*: this public key *may* be valid be we cannot be too sure.
- *complete*: we can be wholly confident that this public key is valid.

3.2 Trustworthiness of introducer

This says how much we can trust a public-key (ie. indirectly referring to the owner of the public-key) to be a competent signer of another PGP public-key certificate. PGP allows the user to assign four levels of trustworthiness to a public-key. These levels correspond to how much the user thinks the owner of this public-key can be trusted to be an 'introducer' to another trustworthy public-key certificate. Trust levels can be one of these:

- *full*: this public-key is fully trusted to introduce another public-key.
- *marginal*: this public-key can be trusted to introduce another public-key, but, it is uncertain whether it is fully competent to do that.
- *untrustworthy*: this public-key should not be trusted to introduce another, therefore any occurrence of this key as a signature on another public-key should be ignored.
- *don't know*: there are no expressions of trust made about this public-key.

The actual meaning of these trust levels are not explicit. It is only prudent to use them as rough guidelines to how much trust to place in an introducer. How the user arrives at her opinion about the introducer's trustworthiness is also left up to the user. However, guidelines on vetting a user's introducer trustworthiness is given in the documents accompanying PGP, including various aspects of the candidate which may affect her credibility as an introducer.

To compensate for the ambiguity of the trust levels, PGP allows its users to tune PGP's 'skepticism'. This is done by adjusting two parameters, COMPLETES_NEEDED and MARGINALS_NEEDED. The former defines the number of completely trusted signatures required to make a certificate completely valid, and the latter defines the number of marginally trusted signatures to achieve the same outcome. A certificate becomes completely valid if *either* one of these skepticism parameters are met. If neither is met, but at least one type (marginal or complete) of signature is present then the signed certificate attains a *marginal* validity status. Since PGP does not explicitly provide mechanisms for expressing security policies, this skepticism mechanism is the closest thing to a policy in PGP [1]. The skepticism level of PGP indirectly reflects the user's own policy regarding the threshold of her confidence in PGP signatures.

We now look back at the example in Figure 2 and assume Carol *fully* trusts Alice and Greg as introducers. If Carol had defined in her PGP environment that at least two fully trustworthy introducers are required to make a certificate completely valid, then Bob's certificate will only be partially valid if only Alice or Greg had signed it. But since both Alice and Greg trusts the validity of Bob's certificate by signing it, Carol can now regard Bob's certificate as completely valid.

However, the skepticism level of PGP is forced to be globally defined throughout the user's own

---

[2]We shall use the term *certificate* to refer to a signed PGP public-key string.

[3]A *valid* public key is a key that is strongly bound to its ID by a fully trustworthy signature on the certificate.

domain. In less technical terms, the skepticism level affects all of the keys in the user's pubic-key ring, no more, and no less. This approach assumes that every public key with the same trust level has exactly the same trustworthiness “value”. In other words each level represent an exact trust value instead of trust groupings to which public keys belong. The first problem is that clearly the limited levels of trust in PGP is insufficient to reflect the highly varying opinions about trustworthiness that a user must put in a public key or introducer. Secondly, in real life each introducer will vary in their trustworthiness with respect to one another. Therefore, in PGP, given two marginally trusted introducers, one of them could be twice more trustworthy than the other. Since the skepticism level in PGP is globally defined instead of on each introducer, treating introducers in this way is not allowed. A more practical approach would be to assign trust points to each trusted introducer, more points for the more trustworthy introducer, then define globally how many points are required to fully certify a public-key certificate.

```
 1. For each signature do

    // scan signatures

 2.   if signature is completely valid then
 3.     if key trust ∈ {undefined,unknown,untrusted} then
 4.        ignore signature
 5.     if key trust is marginal then
 6.        accumulate marginals_counter
 7.     if key trust is complete then
 8.        accumulate completes_counter
 9.   else
10.      ignore signature

    // decision

11.   if (marginals_counter>0) or (completes_counter>0) then
12.     if (marginals_counter>=MARGINALS_NEEDED) or
13.     (completes_counter>=COMPLETES_NEEDED) then
14.        mark key validity as 'complete'
15.     else
16.        mark key validity as 'marginal'
```

Figure 3 How PGP evaluates trust - a high level algorithm. This algorithm represents an observation of how PGP evaluates the certificate, and was not in any way confirmed with the programmers of PGP for its approximation to the real algorithm that was implemented in PGP.

It is appreciated that PGP was never intended to be more than an e-mail encryption software, and trust management issues like this one is assumed to be handled externally to PGP. Furthermore, to add these sort of functionality bulk to PGP would just kill its elegance and attractiveness as a tool to provide ‘encryption for the masses’. A serious user of PGP should understand the implicit trust implications inherent in PGP or any cryptographic authentication and certification tool nonetheless.

4. Evaluating Trust

Only the trustworthiness of a public-key's validity are automatically evaluated by PGP. Introducer trusts are manually assigned by each user to the public keys, and exists only within each individual user’s public-key ring[4]. Public-key validity is also a secret piece of information because it is based on introducer trust (except direct trust). The rationale for this is twofold; firstly to protect each PGP user's personal opinion about other people's trustworthiness, and secondly different people will have potentially different personal opinions about other people's trustworthiness as an introducer. Therefore, introducer trust levels are secret to the user who assigned them.

When PGP evaluates a public-key certificate, it follows roughly the algorithm in Figure 3Figure 3.

Further clarification of the algorithm:

[4]PGP allows users to have files representing multiple ‘key rings’ to store public or secret keys.

line 6: `marginals_counter` accumulates the number of marginally trusted signatures for this key certificate.

line 8: `completes_counter` accumulates the number of completely trusted signatures for this key certificate.

line 10: signatures that are not completely valid are ignored, even when its trustworthiness as an introducer is defined.

lines 11,12,13: MARGINALS_NEEDED and COMPLETES_NEEDED are explained in Section 3.2. If neither minimum-signature requirements are met, but at least one of them totals more than 1[5], then the key is deemed marginally valid (line 16).

5. Introducers

PGP calculates the validity of each public-key based on the signers of that certificate. These signers are called introducers in PGP, ie. one user introduces a new public-key certificate to another user by signing it with its own key. The mechanism for determining a certificate's validity was described in the previous section. When an introducer signs (introduces) a public key certificate, it is signing the statement saying that, as far as she is aware, the public key contained in the certificate indeed belongs to the owner-Id contained in that certificate. The user being introduced to can then make its own judgement on how much to trust that statement based on how much he trusts the introducer.

When a user places trust in an introducer, it implicitly means that the user places a certain amount of confidence in the introducer's capability to introduce valid certificates. In other words, the user trusts the introducer with respect to "introducing correct bindings between a user and her public-key". A marginally trusted introducer may not be as trustworthy as a completely trusted one, therefore more marginally trusted introducers are required to sign a certificate compared to fully a trusted one, for the same level of confidence to be placed in the validity of a certificate. How much trust is placed in any one particular introducer is up to the individual user and is kept secret.

In PGP, for an introduced certificate to be valid, two criteria must be fulfilled. Firstly, the number of introducers must be more or equal to the minimum introducers defined in the PGP skepticism parameters (see previous section). Secondly, the introducers must be directly trusted by the user. The latter becomes a problem when it comes to introducer chains, ie. C can introduce X to A directly but not via B. In both cases A must trust C directly which makes B redundant anyway, but is limiting in terms of disallowing introducer chains of more than one introducer. This also implies that there are no capabilities for introducing other introducers in PGP, ie. there is no mechanism for propagating introducer trust within the PGP web of trust.

Figure 4 Introducer chain**.**

In figure 4, we can see that even though there is a path from Carol leading to Eric via Alice and Bob, Carol cannot immediately assume that Eric's certificate is valid, unless Carol trusts Bob as an introducer, or Alice signs it.

There is however a CERT_DEPTH parameter in PGP which defines the maximum certification chain length, but it is unsure how this is used in evaluating certificate validity. It is suspected that all introducers in the certification chain must be directly trusted by the user. If this is true, then any other introducer, except the last introducer furthest away from the user, in the chain is redundant. Thus, this also makes the CERT_DEPTH parameter redundant in PGP. Further clarification from the creators of PGP on the use of this parameter is required.

6. Completing the example

The complete scenario of the example introduced at the beginning of this article is

---

[5]In other words, $0<n<r$, where n=number accumulated, and r=minimum total. r corresponds to PGP's "skepticism" (see Section 3.2.)

given in below. We'll ignore Eric, Fred and David because they are irrelevant to the interactions that will take place.

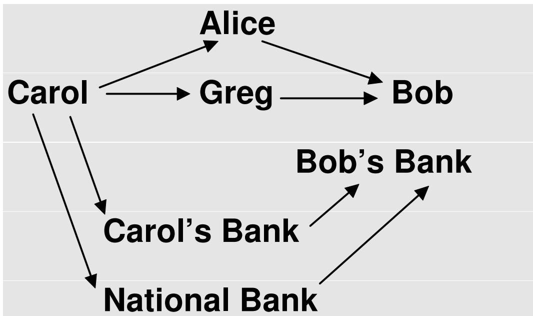

Figure 5 Example subset of an interacting community and its web of trust.

At the current stage, Carol has a copy of Bob's certificate which she finds to be completely valid because it has been signed by two of Carol's fully trusted introducers. Carol can now communicate with Bob and ask for some sample data, before deciding to subscribe to Bob's full service. She does this because she realizes that Bob having a completely valid and trustworthy certificate does not automatically mean that Bob himself is trustworthy enough as a competent/honest/efficient data supplier.

Bob sends Carol a digitally signed sample, and Carol verifies that it is from Bob. Being satisfied with the sample, Carol then applies to Bob for a subscription to Bob's data collection services for her research. Bob then sends Carol the subscription details, and the certificate of his bank which Carol should pay to. Carol finds that Bob's Bank's certificate is signed by Carol's Bank and the National Bank, both of whom Carol trusts fully to be an introducer, so Carol can be confident that Bob's Bank's certificate is completely valid.

Carol then sends Bob's Bank the payment by e-cash together with her public-key certificate which will later be forwarded to Bob so that he can use it to send Carol her data.

7. Conclusion

PGP have been built primarily as an e-mail encryption tool, its scope being setting up a secure channel between two corresponding parties. To assist in forming trust opinions about newly encountered parties, represented by their public-keys, the introducer mechanism is used.

Users are allowed to trust a public-key certificate in two ways, firstly in its validity as a correct binding between the public-key and the claimed owner of that key, and secondly as representing the key of a trusted introducer. Within these two trust "categories" different levels of trust can be expressed, albeit in only 4 levels.

To allow flexibility and to cater for different evaluation policies for different users, PGP's skepticism level can be tuned by defining the minimum number of introducers required for a key certificate to be valid.

Finally, there is no mechanism for propagating trust opinions within the PGP web of trust, therefore introducer chains, or any form of trust related chain, do not exist in PGP.

PGP has been a major leap in providing encryption the public and it has done well within the scope it was designed to work in. However, its use as a more elaborate on-line encryption and authentication tool is not suitable and users have to be aware of the consequences of its trust notions in order to apply PGP to their work routine effectively.

References

[1] J. Feigenbaum, M. Blaze, J. Lacy. Decentralized Trust Management. *Proceedings of the IEEE Conference on Security and Privacy*. Oakland, May 1996.

[2] W. Stallings. The PGP Web of Trust. *BYTE*. February 1995.

[3] P. Zimmermann. *Pretty Good Privacy User's Guide, Volume I and II*. 14 June 1993 Revision. Distributed with the PGP software.