Reasoning about Trust and Insurance in a Public Key Infrastructure

Jonathan K. Millen
SRI International
333 Ravenswood Ave
Menlo Park, CA 94025 USA
millen@csl.sri.com

Rebecca N. Wright
AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932 USA
rwright@research.att.com

April 10, 2000

Abstract

In the real world, insurance is used to mitigate financial risk to individuals in many settings. Similarly, it has been suggested that insurance can be used in distributed systems, and in particular, in authentication procedures, to mitigate individual's risks there. In this paper, we further explore the use of insurance for public-key certificates and other kinds of statements. We also describe an application using threshold cryptography in which insured keys would also have an auditor involved in any transaction using the key, allowing the insurer better control over its liability. We provide a formal yet simple insurance logic that can be used to deduce the amount of insurance associated with statements based on the insurance associated with related statements. Using the logic, we show how trust relationships and insurance can work together to provide confidence.

1 Introduction

Suppose you want to obtain the public key of some person or organization for the usual reasons: to send a private message or to validate a digitally signed message. Unless you can obtain the key directly from a person or organization you recognize, it will be conveyed by a certificate $[A, K_A]_{K_B}$ relating identity information A (a name and address, for example) to the public key K_A, digitally signed so that it can be validated by another public key K_B. For grammatical simplicity, we will say that the certificate is "signed by" K_B even though the key actually used to construct the signature is the corresponding private key K_B^{-1}. In determining whether the certificate is valid, it is necessary both to check the digital signature, and also to determine whether the owner of K_B is trusted to have properly verified the binding between A and K_A before signing the certificate.

Currently, the two main approaches to building a large-scale public-key infrastructure (PKI) are the hierarchical approach, which may be based on X.509 certificates and protocols [AZ98], and the PGP "web of trust" approach [Zim95]. In either model, a participant authenticates user/key bindings by determining one or more paths (sequences) of certificates such that the user trusts the first entity in the path, certificates after the first are signed by the previous entity, and the final certificate contains the user/key binding in question. In both models, the path may be short (perhaps just one certificate) or long. The difference between the two models is in the way trust is conveyed on the path.

In the hierarchical model, a certificate is signed by a certificate authority (CA). Besides a key binding, a CA certificate authorizes a role or privilege for the certified entity, by virtue of its status as an "authority" within its domain. For example, a company can certify its employees' keys, because it hired those employees; a commercial certificate authority (CA) can certify its customer's keys, because it generated them; a government or commercial CA can certify keys of hierarchically subordinate CA's, by its powers of delegation; government agencies can certify keys of government agencies and licensed businesses, as empowered by law; and an international trade bureau can certify government keys, by international agreement. An individual is assumed to know and trust the key of a CA within its domain. A CA is assumed to know and trust the key of the CA who certifies its own keys, and it has a responsibility for accuracy when signing certificates of a principal in its domain. In summary, the hierarchical model conveys trust transitively, but only within a prescribed domain of control and authority.

In the PGP model, individuals act as *introducers*, by certifying the keys of other individuals whom they have personally authenticated. In order for a user A to determine whether a key K_B belongs to a user B, the PGP software considers the signatures certifying the binding of K_B to B (there may be more than one). PGP must ask A whether any of the users who signed B's certificates are considered trusted (completely or marginally) to verify and sign someone else's certificate. In other words, trust is not conveyed along the path of certificates, but rather it is awarded by the user of the certificate. Belief in the final certificate is possible only if the user trusts all of the certifying users on a path.

A limiting factor in the realization of a large-scale PKI has been that the initial authentication of a user-key binding to provide that user with a certificate has had to be done in person. In fact, no large-scale PKI has been fully realized as of date. The hierarchical approach has an apparent advantage of simplicity, but it has been hard to find an organization willing to act as a CA that is both considered trustworthy by many people and has the resources to carry out the initial authentication in person. On the other hand, the PGP approach takes advantage of already existing personal relationships between individuals to solve the problem of in-person authentication. However, if there are not enough users acting as introducers and considered trusted by other users as introducers, then the resulting paths will tend either not to exist or to be long. Unfortunately, long paths provide less assurance because there is more chance that one of the introducers is not, or should not be, trusted.

1.1 The Role of Insurance

In the real world, insurance is used to mitigate financial risk in many settings. For individuals, the fixed moderate cost of paying for insurance is preferable to risking the liability of large sums of money if certain bad events occur. For insurers, risk is pooled and therefore statistically predictable; insurance rates can be adjusted accordingly. Similarly, it has been suggested that insurance can be used in distributed systems, and in particular, in authentication procedures, to mitigate risk there [LMN94, RS99, Ver00].

The use of insurance in a public-key system is not new. Lai, Medvinsky, and Neuman [LMN94] discuss several methods of providing assurance, including liability and surety insurance, of distributed services, including authentication services. They also describe a method of representing and verifying assurance credentials. Reiter and Stubblebine [RS99] further consider the use of insurance-backed public-key certificates and argue that such insurance can be used to provide a better metric of authentication than other methods proposed in the literature. Verisign's NetSure program [Ver00] also provides insurance for some of its public-key certificates.

Two advantages of insurance are that users may be more willing to act as introducers if they do not incur financial risk by doing so, and that longer paths can be useful if the keys involved are insured, even when they do not carry hierarchical authority. Insurance can be used to replace or complement the need for trusted introducers for some or all of the certificates in a path.

1.2 Our Approach

We propose an approach that uses insurance to realize some of the best features of prior approaches. We consider that the insurer of a key may not be the same entity that certifies the user/key binding. This allows us to consider insurers as institutions, while still taking advantage of existing personal relationships to certify user/key bindings. We also suggest the use of threshold cryptography [Des94] to create audited keys, which may make insurers more willing to provide insurance for keys in some circumstances. Furthermore, in order to help reason about insurance of keys and statements signed by them, we provide a formal yet simple insurance logic related to the authentication logic of Lampson, et al. [LABW92].

Our approach differs from Verisign's because it is not hierarchical, and because it can insure the trustworthiness of a key owner as an introducer. Furthermore, unlike Verisign, which only intends its liability to hold in the case that the keys are compromised despite being properly stored, we allow (but do not require) that the insurance may extend to any statement signed by an insured key, not just to certificates. Our approach differs from the Reiter-Stubblebine metric because of the separation between insurer and introducer and because of the insurance logic. Our proposal also differs from that of [RS99] in that we do not require all keys in a set of paths to be insured in order to reason about the degree of assurance in the target key.

The LABW logic [LABW92] characterizes the "speaks for" relation. A public key speaks for its owner in the sense that if the key has not been compromised, only the owner is able to use the key to sign statements. Our logic introduces the "insured by" relation, which allows deduction of the insurance associated with statements. In the case that an injured party wishes to obtain payment for damages, derivations obtained by the logic can indicate which parties are liable.

To summarize, the contributions of this paper are the following.

- We present a method for using insured public keys to facilitate the creation of a large-scale public-key infrastructure.
- We describe an application using threshold cryptography in which insured keys

have an associated auditor who is involved in any creating any signature using the key, thereby allowing the insurer more control over its liability.

- We provide a formal yet simple insurance logic that can be used to reason about the insurance of keys and statements.

- Using our logic, we show how to analyze several examples in which insured keys are used. The results can be used to demonstrate the insurance of various statements, and also to help determine who is liable if those statements turn out to be false.

We present our insurance proposal in Section 2. We describe the use of audited keys in Section 3. In Section 4, we describe the insurance logic and show some examples of its use. We conclude in Section 5.

2 Insurance in a Public-Key Infrastructure

In our proposed approach, accredited insurers can provide insurance for cryptographic keys. Insurers provide certificates stating the insurance relationship. An *insurance* certificate, written $[\$Z, K, a]$, indicates that K is insured by Z for up to a dollars. We will also have the usual user/key binding certificates, which we call *binding certificates* to distinguish them from insurance certificates. An insurance certificate, like binding certificates, normally carries a digital signature, as $[\$Z, K, a]_{K'}$. Z may or may not be the owner of the key K' with which the certificate is signed. In particular, if the user of the certificate does not know the insurer, it may be more useful to have a certificate signed by someone else. If the signer's key is insured or introduced by another certificate, this creates a path of certificates.

An insurance certificate has certain contractual and legal obligations. Roughly speaking, if a statement signed by an insured key turns out to be false, then the insurer may be liable. Whether or not the insurer is liable depends on the specific terms of the policy. In reality, as with any insurance settlement, there may be a complicated process to determine exactly how much is paid, by whom, and to whom. However, there are already procedures in place for current insurance practices that can be adapted for this new setting.

As with current insurance practice, it will not always be possible to correctly determine precisely the events that have occurred, due to incomplete, incorrect, or misleading information, but there is a delicate but workable balance between many factors to help stabilize the infrastructure into one that can be maintained in a practical way. Insurance rates are tailored to allow insurers to profit even if they must pay some settlements only because they could not prove that they were not liable. Large-scale or systematic misbehavior, either by insurers or by insured parties, is likely to be caught and punished. For insurers, factors such as government regulations, court judgements, and public perception work to help ensure that they pay when required to do so. For the insured, the possibility that settlements will not be paid if fraud is detected, as well as the threat of criminal penalties, help prevent fraudulent behavior.

It is unlikely that insurers will want to take on the risk of liability for all possible uses of digital signatures. Insurance policies can state restrictions on what kind of uses are allowed for particular keys. If desired, insurers can require insured keys to be audited, allowing the auditor control over which signatures will be allowed. This possibility is discussed further in Section 3.

A natural and useful restriction, on which we will focus most of our attention, is to consider only the use of keys to sign public-key certificates. In particular, we propose a public-key infrastructure where an individual's keys are certified by other individuals acting as introducers, as in PGP. The trustworthiness of an introducer is replaced by insurance.

At this point, it is useful to consider a simple example. Suppose that we have a binding certificate $[A, K_A]_{K_B}$ and an insurance certificate $[\$Z, K_B, a]_{K_Z}$, and that we know (somehow) that K_Z is Z's public key. Here, A is a user whose public key is being certified, Z is an insurer, and B is the introducer who has a policy with Z. Ordinarily Z would know who B is. (If B is anonymous, Z would have to accept responsibility that would be passed to B in some cases.) Suppose further that K_A is later found to have signed a statement $S =$ "A owes C the sum of \$100," and A refuses to pay. C's belief that K_A was A's signature was based on the certificate signed by K_B.

If A agrees that she signed S, then there is no reason to hold Z liable. In this case, C should take A to court to try to obtain payment for the statement S signed by A. However, if A repudiates S, saying that she never signed it, this is a claim by A that the binding between K_A and A, which was signed by the insured key K_B, is incorrect. In this case, C should try to collect from Z. There are a number of different ways that this latter case could arise. In general, C's filing a claim due to repudiation will result in an investigation by Z to attempt determine which of the cases below has occurred, and to take appropriate action:

- A is lying now. This constitutes fraud on the part of A, and is a crime. In this case, logically, Z should not pay; it is just a matter for B and Z to get together and present the evidence that A's certificate was valid.

- K_A is not actually A's key, because B, either intentionally or unintentionally, did not properly authenticate A before signing the certificate. Depending on the specifics of the policy between Z and B, and whether B can show any records of having followed some kind of proper procedure in authenticating A before signing A's certificate, Z or B is liable. If Z and B disagree about who is liable, a court judgement may be needed.

- K_A has been compromised. Even though B correctly identified A as the owner of K_A, Z might still be liable if K_A had been publicly revoked and B had failed at the time to check the applicable revocation list. Auditing information such as timestamps in certificates may be needed to help determine whether the key binding in question was correct in the sense covered by Z.

- K_B has been compromised; an attacker forged the signature on A's certificate. Since K_B is insured by Z and has been compromised, Z should pay. (Note that this case is the only case that the Verisign NetSure model insures against.) However, if it can be shown that B did not properly safeguard K_B, then Z may decide that B violated the terms of the contract, and Z will inform C

to go after B. Whether B is seen to be liable by a court depends on the circumstances under which digital signatures are held to be binding.

- K_Z has been compromised or was improperly identified as Z's key, or Z is not an accredited insurer. This should be extremely unlikely as insurers' keys are assumed to be well-known and well-protected, and similarly it should be easy to determine whether an entity is an accredited insurer. In the case that K_Z has been compromised, it might be determined that either Z or some kind of fund to which all insurers contribute should pay.

As even this small example illustrates, it can be complicated to determine who is liable when a dispute arises. In Section 4, we present a formal logic that can be used to identify a list of possibly liable parties. We will revisit this example there. However, many of the practical real-world details are necessarily outside the scope of the logic.

3 Audited Keys

Because statements signed with insured keys are themselves insured, insurers may wish to place some restrictions on the kinds of statements that are signed with insured keys. There are several methods by which this can be achieved, which would most likely work in combination. Courts can be relied on to define and recognize what statements are considered contracts. Alternately, the policy between a user A and the insurer Z can spell out exactly what kinds of statements A is allowed to sign with a key K_A, or what kinds of signed statements are insured. Similarly, the standard notions of what constitutes a contract and who can be considered an injured party will apply.

A novel method for an insurer to enforce such restrictions is to require, using a threshold signature scheme, that an auditor (who is possibly the same entity as the insurer) participate in every signature. (A survey of threshold cryptography can be found in [Des94]; some threshold signature schemes are presented in [DF91, GJKR96].) In two-out-of-two threshold signature schemes, two parties hold *shares* of a private key K^{-1}. Computing signatures with K^{-1} requires participation of both parties; neither party can compute signatures without the help of the other. The resulting signature can, as usual, be verified by anyone using the corresponding public key K.

Two-out-of-two threshold signature schemes can be used for auditing as follows. A user A, whose public key is K_A, has one share of the corresponding private key K_A^{-1}. The auditor holds the other share. These shares can be generated by a trusted third party or can be generated by A and the auditor themselves [GJKR96, BF97, FMY98]. Note that neither A nor the auditor learns the entire private key, but both learn the corresponding public key K_A. Assuming neither of the shares of the private key nor the entire private key are compromised[1], A cannot produce signed documents without the involvement of the auditor. Similarly, the auditor

[1]Even though the entire private key need not be known to any of the participants, it could still be compromised, for example in the case of RSA, by an attacker who is able to factor the public modulus.

cannot forge A's signature. Additionally, other users, with or without the help of the auditor, cannot forge A's signature.

Now suppose that an insurer Z insures K_A, but wishes to restrict the use of K_A to sign only particular kinds of statements. Each time A wishes to sign a statement, he must communicate (electronically) with the auditor, who will verify that the statement being signed is of the proper form before participating in creating the signature. For example, to use auditing with keys intended only for the purpose of creating public-key certificates, the auditor would verify that the statement being signed was in fact a certificate. If desired, the auditor could also verify other properties, such as that the key length is long enough to be considered secure.

Note that recipients of signatures need not know whether a key is being used auditably, even in order to verify the signature. That is, A can still be given only the insurance certificate $[\$Z, K_B, a]_{K_Z}$ showing that the public key K_B is insured by Z. If desired, special auditor certificates can also be introduced to indicate an auditing relationship, but this is not necessary.

The use of auditors gives insurers more control over their liability, and also can be used to ensure that users are meeting the terms of their policies. Because the auditor can refuse to participate in the signing of statements that do not meet the insurer's requirements, auditing potentially reduces the insurer's risk (of having to pay large settlements on certain statements) and court costs (of having to prove that they are not liable for certain statements) by controlling which statements insured keys sign.

Another use of auditing is that the auditor can check that the user is up to date on premium payments before participating in creating a signature. In addition, the auditor can keep records of which statements were signed with which keys. These records could be useful for checking that an insured key is not being used more than allowed by the insurance policy, or to detect certain kinds of compromised keys. For example, if K_A is an audited key and a claimant presents a statement signed by K_A that is not in the auditor's record, then this implies that either the entire private key K_A^{-1} or the auditor's share of it has been compromised, or the auditor's records themselves have been compromised.

Auditing can also provide benefits to the user. For example, an insurer may be willing to give a policy for audited keys that indemnifies the user completely, since the auditor can refuse to sign "risky" statements when asked. Insurers may chooses to require auditors for all users, or only for some users, or may offer lower insurance premiums for audited keys than for non-audited keys.

4 Insurance Logic

In this section, we describe a method for reasoning about insured keys to derive insurance about the statements signed by those keys. Specifically, we extend the delegation logic of Lampson et al [LABW92] to handle insurance by adding three axioms.

The LABW logic interprets a certificate as a statement that a key "says" some statement implied by the format of the certificate. Given an additional statement that the key "speaks for" a principal, one concludes that the principal said (i.e. uttered, believed, authorized) the same statement. Let $K \Rightarrow A$ be an abbreviation for "K speaks for A." Informally, this means that K is a public key owned by A. It should also mean that A is responsible for statements (such as contracts) that are signed with K. Thus, if K says something, we may act as though A said it. The formal meaning of "speaking for" is embodied in the following axiom:

A1. If $K \Rightarrow A$ and K *says* S then A *says* S

In the LABW logic, A1 is not an axiom, but rather a consequence of a definition of $\Rightarrow$ involving the compound principal $A \wedge K$. In our application, we will not need compound principals, so for simplicity we take this directly as an axiom.

A binding certificate $[A, K_A]_{K_B}$ is interpreted in the logic as the statement K_B *says* $K_A \Rightarrow A$. An insurance certificate $[\$Z, K_Z, a]_{K_B}$ is interpreted in our logic as K_B *says* $K_Z \, \$_a \, Z$.

We add three axioms that represent the properties of insurance. The first property of an insured key is that statements signed by the key are also insured:

A2. If $K \, \$_a \, Z$ and K *says* S then $S \, \$_a \, Z$

Second, a principal can commit itself to a liability:

A3. If Z *says* $P \, \$_a \, Z$ then $P \, \$_a \, Z$

In A3, P can either be a statement or a key.

Finally, the falsity of an insured statement creates a liability for the insurer provided that the terms of the insurance are met. We write $Z \, \$ \, a$ to mean that Z is liable for the amount a provided that the terms of the associated insurance are met. The liability axiom is then:

A4. If $S \, \$_a \, Z$ and $\neg S$ then $Z \, \$ \, a$

As discussed in Section 2, this does not necessarily mean that Z is necessarily liable if S is false, but rather that Z is a reasonable entity to go after when seeking damages caused by S being false. In practice it means that Z is liable unless Z can show another party is liable instead. As we will see in the following examples, we can use our logic to derive liability statements.

4.1 A Simple Example

At this stage, we can give an example of a useful deduction. We return to the example discussed at the end of Section 2. Suppose we have certificates $[A, K_A]_{K_B}$ and $[\$Z, K_B, a]_{K_Z}$, and suppose further that we can assume that K_Z is Z's public key. The latter assumption is interpreted in our logic as the statement:

(1) $K_Z \Rightarrow Z$

The certificates are interpreted as:

(2) K_B *says* $K_A \Rightarrow A$

(3) K_Z *says* $K_B \, \$_a \, Z$

A helpful visual representation is to create a diagram as follows:

$$(K_Z \Rightarrow Z) \rightarrow (K_B \,\$_a\, Z) \rightarrow (K_A \Rightarrow A)$$

where an arrow is a *says* relation using the key mentioned on its left.

From (1) and (3),

(4) Z *says* $K_B \,\$_a\, Z$

From (4) and A3 we have

(5) $K_B \,\$_a\, Z$

and by A2, (2), and (5), we have

(6) $(K_A \Rightarrow A) \,\$_a\, Z$

That is, we are able to conclude as desired that the link between A and K_A has been insured by Z. Note that it is not possible to derive $K_A \,\$_a\, Z$. That is, our logic upholds the desirable property that the fact that B, who is insured by Z, acts as an introducer for K_A does *not* imply that Z is liable for statements signed by K_A, but only for the binding between A and K_A.

Again continuing with our example from Section 2, suppose that K_A is later found to have signed a statement $S =$ "A owes C the sum of \$100," and A refuses to pay. As we argued there, there are several ways this could happen that do not create a liability for Z. In terms of our logic, that is because, in those cases, the statement $K_A \Rightarrow A$ is true, so (6) does not apply. However, if B did not properly authenticate A, or if K_A has been compromised, then:

(7) $\neg(K_A \Rightarrow A)$

In this case, it follows from (7) and Axiom A4 that:

(8) $Z \,\$\, a$

That is, C can expect to recover those damages, up to the amount of a, from Z, provided the terms of the insurance are met.

4.2 Longer Paths

In this section, we further demonstrate the utility of the logic by briefly considering some slightly more complicated examples. We first show how a path of insurance certificates can be coalesced into a single insurance statement. Following that, we analyze a path in which some certificates are insured and some are trusted.

Multiple insurers along a path For this example, suppose that K_B is insured by Y, but the insurance certificate we have for K_B is signed by a key K_D not known to be Y's key. Additionally, K_D is insured by Z, who has signed an insurance certificate. That is, we are assuming:

(1) K_D *says* $K_B \,\$_a\, Y$

(2) K_Z *says* $K_D \,\$_b\, Z$

(3) $K_Z \Rightarrow Z$.

This would be diagrammed as

$$(K_Z \Rightarrow Z) \rightarrow (K_D \,\$_b\, Z) \rightarrow (K_B \,\$_a\, Y)$$

We conclude in a few steps that

(4) $(K_B \,\$_b\, Y) \,\$_a\, Z$

Again, we ask what happens if a statement signed by K_B is false. That depends on whether $K_B \,\$_b\, Y$, as determined by some factual investigation. If true, the liability axiom says $Y \,\$\, b$. If false, we have $Z \,\$\, a$. In either case, an injured party seeking damages can hope to collect at least the smaller of a and b. This sort of case analysis can be extended to consider multiple independent paths, leading to the min-cut metric proposed by [RS99].

Combining trust and insurance in a path Here, we extend the example from Section 4.1 by adding the additional assumptions that A is trusted as an introducer, and that we have a certificate $[D, K_D]_{K_A}$[2]. These assumptions are represented as follows:

(1) If A *says* $(K_D \Rightarrow D)$ then $K_D \Rightarrow D$

(2) K_A *says* $(K_D \Rightarrow D)$

In addition, we have all the assumptions and conclusions from Section 4.1. The extended diagram is as follows:

$$(K_Z \Rightarrow Z) \rightarrow (K_B \,\$_a\, Z) \rightarrow (K_A \Rightarrow A) \rightarrow (K_D \Rightarrow D)$$

Note that we are assuming that the entity A is trustworthy, rather than a key. This assumption is useful for demonstrational purposes, but also reflects that fact that trust is usually based a personal relationship with a user, rather than with a key. The user/key binding between A and K_A is not directly assumed, but it can be derived that the binding is insured by Z. In particular, recall the conclusion reached in Section 4.1:

(3) $(K_A \Rightarrow A) \,\$_a\, Z$

Hence, by A4, it follows that either

(4) $K_A \Rightarrow A$, or

(5) $Z \,\$\, a$

[2] For simplicity, we state here only the assumption that A is trusted to introduce the key and user in this particular certificate. A fully general treatment would use a universal quantifier and then infer the particular statement needed.

Supposing that (4) is true, in a few steps one can derive:

(6) $K_D \Rightarrow D$

Suppose that now a user C shows damages based on a document signed by K_D (and by beliefs in these assumptions). If it is determined that $\neg K_D \Rightarrow D$, then (6) is false. If further investigation determines that in fact A is not trustworthy, then the trust in A as an introducer was misplaced, and it is a personal decision whether C wants to try to recover damages from A or not. However, if it is determined that in fact the problem was that (4) is false, i.e. $\neg(K_A \Rightarrow A)$, then it follows from (5) that Z should pay.

5 Discussion

Our PKI approach is most useful in an environment where certificates can be signed by introducers other than certification authorities whose trustedness is beyond question. It is more formal than the PGP "web of trust" in which individuals must make their own unsupported decisions as to the trustedness of introducers, yet it also allows users to incorporate their own beliefs about who are trustworthy introducers into their decisions if desired. Furthermore, it is not necessary to have a hierarchy of insured introducers—any path will do—and not all introducers on the path need be insured. We imagine that, as is very common with PGP, some users will certify each other's keys; such users may or may not choose to have their own keys insured. In addition, some users will act as notaries public; these users will certify more keys, and would be expected to have their own keys insured.

The role of insurers is important. Insurers' keys are intended to be more well-known and well-protected than regular user keys. We envision a world in which there are a fairly small number of insurers. Insurance keys are assumed to be easy to verify. While we do not require all users to know all insurers' keys, we think it is reasonable to assume that insurers can determine other insurers' keys, and that each user knows at least one insurer's key. Insurers' keys are extremely valuable targets and should therefore be properly protected; the ability to do this should be one of the requirements of being accredited as an insurer. Similarly, unscrupulous insurers who misbehave too frequently or too severely will be detected and punished.

There are several advantages to this PKI approach. Like PGP, we can use existing personal relationships to perform certifications. However, because of the insurance, we do not require the user to personally trust all introducers in a path. Furthermore, we believe our approach is easily implemented because most people already have a relationship with some insurance company; extending existing insurance business models to cover this case should be easier than starting new CAs. Another advantage is that not all introducers need be insured: a user may still choose to trust some other users as introducers even without insurance. In that case, provided that their assumptions about trusted introducers are correct, any incorrect key-binding pairs will involve an insured introducer. In addition, note that some risks can be lessened by taking certain precautions. Insurers can encourage users to take such precautions by charging lower premiums to customers who agree to adhere to risk-reducing behaviors. (Many car insurers have lower premiums for seat belt wearers.) For example, keys are less likely to be compromised if they are chosen keys properly (i.e. large enough and randomly) and properly safeguarded. In order to obtain lower premiums, many users may adopt these risk-reducing behaviors, which has the beneficial side effect of providing better security for everyone.

References

[AZ98] C. Adams and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Certification Server Protocols," Internet Draft, PKIZ Working Group, 1998.

[BF97] D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," In *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 424–439, Springer-Verlag, 1997.

[Des94] Y. Desmedt, "Threshold cryptography," *European Transactions on Telecommunications and Related Technologies*, 5(4):35–43, July–August 1994.

[DF91] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures," In *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 457–469, Springer-Verlag, 1992.

[FMY98] Y. Frankel, P. MacKenzie, and M. Yung, "Robust efficient distributed RSA key generation," In *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pages 663–672, May 1998.

[GJKR96] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust Threshold DSS Signatures," In *Advances in Cryptology—CRYPTO '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 354–371, Springer-Verlag, 1996.

[LMN94] C. Lai, G. Medvinsky, and B. C. Neuman, "Endorsements, Licensing, and Insurance for Distributed System Services," In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pp. 170–175, November 1994.

[LABW92] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in Distributed Systems: Theory and Practice," In *ACM Transactions on Computer Systems*, Vol. 10, No. 4, pp. 265–310, November, 1992.

[RS99] M. K. Reiter and S. G. Stubblebine, "Authentication Metric Analysis and Design," In *ACM Transactions on Information and System Security*, Vol. 2, No. 2, pp. 138–158, May 1999.

[Ver00] Verisign, `https://www.verisign.com/netsure/index.html`.

[Zim95] P. Zimmermann, *The Official PGP User's Guide,* MIT Press, 1995.