Confidence Valuation in a Public-Key Infrastructure based on Uncertain Evidence[*]

Reto Kohlas Ueli Maurer

Department of Computer Science
Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland
{kohlas,maurer}@inf.ethz.ch

Abstract. Public-key authentication based on public-key certificates is a special case of the general problem of verifying a hypothesis (that a public key is authentic), given certain pieces of evidence. Beginning with PGP, several authors have pointed out that trust is often an uncertain piece of evidence and have proposed ad hoc methods, sometimes referred to as trust management, for dealing with this kind of uncertainty. These approaches can lead to counter-intuitive conclusions as is demonstrated with examples in the PGP trust management. For instance, an introducer marginally trusted by a user can make him accept an arbitrary key for any other user.

In this paper we take a general approach to public-key authentication based on uncertain evidence, where not only trust, but also other pieces of evidence (e.g. entity authentication) can be uncertain. First, we formalize the assignment and the valuation of confidence values in the general context of reasoning based on uncertain evidence. Second, we propose a set of principles for sound confidence valuation. Third, we analyze PGP and some other previous methods for dealing with uncertainty in the light of our principles.

Key words. Public-key certification, public-key infrastructure (PKI), web of trust, Pretty Good Privacy (PGP), evidence theory, reasoning with uncertainty.

1 Introduction

1.1 Motivation

Public-key cryptography is a basic technology for information security and electronic commerce. A prerequisite for the application of public-key cryptography is that the keys are authenticated. The validation of public keys is hence of paramount importance.

This is achieved by public-key certificates. A certificate is a digitally signed statement by which a certification authority (e.g., called trusted third party or introducer) asserts that a key is bound to an entity. The term "binding" is

[*] In the proceedings of the *International Workshop on Practice and Theory in Public-Key Cryptography 2000, PKC2000, Lecture Notes of Computer Science, Springer.*

ambiguous [8] but for the purpose of this paper this is not relevant. The term public-key infrastructure (PKI) is used to refer to the complete legal, technical and organizational framework for drawing conclusions from a given set of certificates, trust relations and other pieces of evidence.

A collection of certificates where for instance Bob certifies Carol's key and Carol certifies Dave's key (and so on) is called a *certification path*. Since a chain is at most as strong as the weakest link, PGP [20, 22] introduced the use of parallel certification paths in order to improve the reliability of a decision about the authenticity of a public key. A certification structure with multiple certification paths is sometimes referred to as a *web of trust*.

One of the uncertainty factors in public-key authentication is the possible intentional or unintentional misbehavior of entities, either by not carefully authenticating a person before issuing a certificate, or by creating false certificates in the first place. Both trust and authentication are hence uncertain to some degree and should preferably be modeled as such. A user (say Alice) should be able to express that she considers one entity more trustworthy than another, or that she considers authentication by passports more secure than authentication by voice recognition over a telephone line.

A value assigned to a piece of evidence as a degree of confidence is called a *confidence value*. A confidence value can be a discrete numerical or non-numerical value, or it can be a real value which may or may not be interpreted as a probability. The problem of assigning and evaluating confidence values numerically or on a discrete scale (as in PGP) is non-trivial, in particular when certification paths intersect, as will be illustrated.

In PGP 2.6.2 for example, a user Alice assigns a value from the set {`unknown, no trust, marginally trusted, fully trusted`} to every key K she retrieved from the PKI. This trust value for K implicitly stands for her trust in the entity that presumably controls K. All keys generated by Alice herself are by default authentic (or `valid`). To determine the validity of a key K, only the signatures under K generated with `valid` keys are considered. A key is accepted to be `valid` if it is signed at least by one `fully trusted` key, or by two `marginally trusted` keys.[1]

Figure 1 shows two PGP-like webs of trust in a graphical notation introduced by Stallings [20]. A circle stands for an entity-key pair, and an arrow from A to B means that A's public key has been signed by B's public key. In the left scenario of Figure 1 for instance, X_1 has issued a certificate asserting that a certain public key is controlled by X_3. Different patterns indicate the different trust values that an entity (in our examples Alice) assigned to a key. In the left scenario for instance, Alice `fully trusts` X_1 and X_2, and she marginally trusts X_3 and X_4.

In PGP 2.6.2, in the left scenario Bob's key is accepted to be `valid` while it is not in the right scenario. Although in the right scenario X_3 and X_4 are

[1] This is a simplified description of PGP's trust management. For example, Alice can choose an upper bound for the length of certification paths to be considered. In the newer PGP releases, another trust management is implemented.

fully trusted, Bob's key is not valid, since already the keys X_3 and X_4 are not valid. One can argue that the two scenarios are isomorphic in a certain sense and that therefore it is counter-intuitive that the validity of Bob's key is different. In both scenarios, the same coalitions of misbehaving entities can cause Alice to accept a false key for Bob: any one of the sets $\{X_1, X_2\}$, $\{X_1, X_3\}$, $\{X_2, X_4\}$ and $\{X_3, X_4\}$ (or of course a superset thereof). The two cases are isomorphic in the sense that in each case, one of the sets consists of two fully trusted entities, two sets consist of one fully trusted and one marginally trusted entity, and one set consist of two marginally trusted entities. It is one of the goals of this paper to formalize such principles like the described isomorphism.

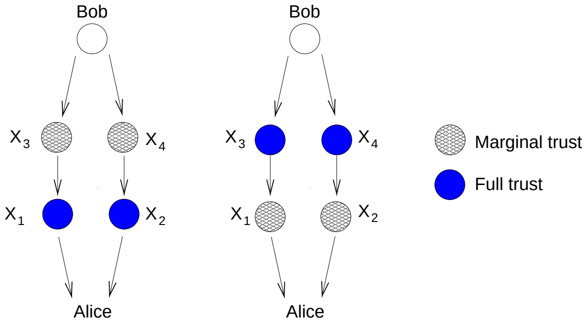

Fig. 1. Examples of public-key rings in PGP.

1.2 Contributions of this Paper

In this paper we take a general approach to public-key authentication based on uncertain evidence, where not only trust, but also other pieces of evidence (e.g. entity authentication) can be uncertain. Two papers with similar goals are [11] and [17].

Generic techniques for reasoning with uncertainty have been proposed in artificial intelligence (e.g., automated medical diagnosis) and decision theory (see, for instance, [13, 19, 9, 12, 6, 10]). Many techniques and approaches for reasoning with uncertain information are based on probability theory [13, 12, 6]. There are at least two conceptually different approaches to using probabilities in this context. In the first approach, probabilities are related to measured or estimated relative frequencies of an event under consideration (e.g. the relative frequency of a disease, given the particular evidence and symptoms observed on a patient), and such systems are often called expert systems.

In the second approach, probabilities are used as parameters of subjective belief, without necessarily having a direct interpretation as relative frequencies.[2]

[2] In fact, the initial motivation (e.g. by Bernoulli [2]) for introducing probabilities was to argue about degrees of belief, in particular in the context of weighting statements

For instance, if one assigns a trust value of, say, 80% to a person, this generally does not mean that one expects the person to misbehave 20% of the time. Nevertheless, interpreting this parameter as a probability of 0.8 makes sense. This is a point that is often misunderstood [3, 16, 14]. Of course, the parameters are generally based also on some form of past experience, but such experience almost never arises in the form of relative frequencies. Moreover, while confidence parameters in expert systems can often be verified in the real world, this is not the case for this second type of probability-based parameters.

Hence we disagree at a conceptual level with the approach taken by Beth, Borcherding and Klein [3] and principle 2 of Reiter and Stubblebine [16, 14]: it is not necessary to define the parameters (i.e., the confidence values) of an authentication method as "negative and positive experiences" or frequencies.

We formalize the assignment and the valuation of confidence values in the general context of reasoning based on uncertain evidence. From an abstract point of view, in any uncertainty method, an entity assigns confidence values to the pieces of evidence she collected; these confidence values stand for her degree of belief that the corresponding piece of evidence is true. We then propose a set of principles for sound confidence valuation. These principles describe how a confidence valuation should reduce the confidence values assigned to the pieces of evidence to a single confidence value for the hypothesis under consideration. Two key concepts in the characterization of the confidence valuation are the notions of assumptions and arguments. These concepts are borrowed from the so-called argumentation systems [7, 5]. Finally, we analyze PGP and some other previous methods for dealing with uncertainty in the light of our principles.

While the main contribution of the paper is on modeling uncertainty, we also make some observations regarding the modeling of evidence in the context of PKI's which are valid regardless of whether one considers evidence to be uncertain.

1.3 Previous Work and Outline

The design of methods for valuating the authenticity of public keys has received substantial attention in the cryptographic community (see [21, 22, 3, 15, 16, 11, 17, 14]); most of these methods are ad hoc and will not be discussed here in detail. Reiter and Stubblebine stated guidelines for a "metric" of authentication ([16]) which will be discussed in the concluding section.

In Section 2 we discuss various ways of modeling and dealing with uncertainty. Key concepts of propositional logic and argumentation systems are revisited. We formalize the concept of an uncertainty method by introducing the notions of confidence value, confidence assignment and confidence valuation. In Section 3 we state desirable principles for confidence valuation. In Section 4 we analyze existing confidence valuation methods in the light of our principles. We show some problems arising in PGP's method for combining trust values. Finally, in

made by different witnesses, where one cannot define an experiment in which relative frequencies make sense.

Section 5, we compare our work with the principles of Reiter-Stubblebine [16] and mention some directions for future research.

2 Reasoning with Uncertainty

A *hypothesis* h is a statement for which one generally does not know whether it is true or not. In order to evaluate the truth of h, one can exploit dependencies of h on other facts whose truth one can observe or estimate. Such other facts or observations are often called *pieces of evidence* or simply *evidence.*

2.1 Propositional Logic and Logical Consequence

Logic is about relating the truth of different units of concern to each other, and hence logic allows to describe the truth of a hypothesis in dependency of a user's (Alice's) evidence. In this paper, we will use propositional logic, but concepts such as logical consequence, assumptions and arguments could also be defined in the context of more powerful languages, for instance first-order logic (for an excellent introduction to different logics see [1]).

The basic units of concern in propositional logic are called *propositions* or, alternatively, *statements.* We denote the set of statements by $\mathcal{S}$, and statements by s,s_1, ... The statement standing for the hypothesis is sometimes denoted by h. A *formula* of propositional logic is composed of statements and *logical connectives.* In the sequel, let $g, f, f_1, f_2, \dots$ stand for formulas. In a standard definition of propositional logic, there are three connectives: $\neg, \wedge, \vee$. "Implies" $(f \rightarrow g)$ is a shorthand for $\neg f \vee g$.

A formula is either true or false, depending on the truth values that have been assigned to the statements. A *truth assignment* $\mathcal{B}$ is a function from the set of statements $\mathcal{S}$ to the set of truth values, $\{$*true*, *false*$\}$, often represented by 1 and 0: $\mathcal{B} : \mathcal{S} \rightarrow \{0, 1\}$. The semantic definition of propositional logic states how the different logical connectives combine the truth of the corresponding subformulas, i.e., what the truth value of a formula f, denoted by $\hat{\mathcal{B}}(f)$, is. The logical connective $\neg$ stands for "not": the formula $\neg f$ is true only if f is false. The formula $f \wedge g$ is true if f is true *and* g is true, and $f \vee g$ is true if f is true *or* g is true. A truth assignment $\mathcal{B}$ such that the formula f is true (i.e., $\hat{\mathcal{B}}(f) = 1$) is called a *model* for f. A formula that has at least one model is called *satisfiable*, and otherwise *unsatisfiable.* A formula g is *logical consequence* of f (or f follows from g), if every model of f is also model of g. This is denoted by $f \models g$. If f and g have the same set of models, i.e. $f \models g$ and $g \models f$, then f and g are called *semantically equivalent.*

One can represent each piece of evidence and the hypothesis by a statement in $\mathcal{S}$, and one's belief is a formula Σ over $\mathcal{S}$. The hypothesis h is accepted if, whenever Σ is true, also h is true. This corresponds to $\Sigma \models h$.

2.2 Assumptions and Arguments

Pieces of evidence can also be uncertain, and not only the hypothesis. If a hypothesis h does not follow from an initial belief Σ there are sometimes assumptions one can make about the truth values of some uncertain pieces of evidence in order to derive h. Informally, such a combination of assumptions is called an argument for h. Often there are different arguments for h; the set of arguments is what one could call a qualitative characterization for the uncertainty of h.

The notions of assumptions and arguments as used in this paper have been introduced in the context of assumption-based truth maintenance systems (ATMS) [4], and formalized in the context of argumentation systems [5]. We will define assumptions and arguments similarly as it has been done in the case of argumentation systems. However, since we do not need the entire power of argumentation systems, we can use a simpler notation.

An *assumption* is a piece of evidence; we denote the set of pieces of evidence by $\mathcal{E}$, where $\mathcal{E}\subseteq \mathcal{S}$. The pieces of evidence, i.e. the assumptions are denoted by $a, a_1, \ldots$

A *conjunction* is of the form $l_1 \wedge \ldots \wedge l_m$, where the l_i are literals; a literal is a propositional statement s or its negation $\neg s$. A conjunction is non-contradicting, if no statement s occurs positively and negatively in the conjunction. In the sequel, let $\mathcal{L}_A$ denote the set of literals over the set of assumptions $\mathcal{E}$ and $\mathcal{C}_A$ denote the set of non-contradicting conjunctions over $\mathcal{E}$. We write a conjunction $l_1 \wedge \ldots \wedge l_m$ also as a set $\{l_1, \ldots l_m\} \subseteq \mathcal{L}_A$.

An *argument* $\mathcal{A}$ for a hypothesis h is a non-contradicting conjunction over the set of assumptions $\mathcal{E}$ (i.e, $\mathcal{A} \in \mathcal{C}_A$) such that h can be derived from $\mathcal{A}\wedge\Sigma$: $\mathcal{A}\wedge\Sigma \models h$.[3] In fact, we could have also defined that an argument is any formula over $\mathcal{E}$. However, as shown below, the arguments for h with respect to Σ can always be represented by a set of so-called minimal non-contradicting conjunctions. Such a set of conjunctions for h is called an *argument structure* and is denoted by $\mathcal{A}^*$, where $\mathcal{A}^* \subseteq \mathcal{C}_A$.

Let $\mathcal{A}$ be any propositional formula over $\mathcal{E}$, such that $\mathcal{A} \wedge \Sigma \models h$. $\mathcal{A}$ is semantically equivalent to a formula $\mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$, where the $\mathcal{A}_i$ are conjunctions in $\mathcal{C}_A$. Every $\mathcal{A}_i$ is also an argument for h, since the set of models for $\mathcal{A}_i$ is contained in the set of models of $\mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$. Furthermore, if the conjunction $\mathcal{A}_i = a_{i1} \wedge \ldots \wedge a_{im}$ is argument for h, then so is the conjunction that is obtained by adding one assumption to $\mathcal{A}_i$. A formula $\mathcal{A} \in \mathcal{C}_A$ such that $\mathcal{A}$ is satisfiable and $\mathcal{A} \wedge \Sigma$ is unsatisfiable is called an argument for the *contradiction.*

2.3 Argument Structures for Horn Formulas

In this paper, we investigate the special case where Σ is a so called *Horn formula.* A *Horn formula* is a conjunction $f_1 \wedge \ldots \wedge f_n$ of *Horn clauses* f_i. A *Horn clause*

[3] In the literature, such arguments are called *supporting* [5]. An argument $\mathcal{A}_2$ such that h is still possible, i.e. such that the counter-hypothesis $\neg h$ cannot be derived ($\mathcal{A}_2 \wedge \Sigma \not\models h$), is called a *plausible* argument.

f_i is a formula $s_1 \wedge \ldots \wedge s_n \rightarrow s_{n+1}$, where the statements $s_1, \ldots, s_n$ are called the *preconditions* and s_{n+1} the *postcondition*. A Horn clause $s_1 \wedge \ldots \wedge s_n \rightarrow s_{n+1}$ means that the statement s_{n+1} must necessarily be true if $s_1 \ldots s_n$ are true.

In the following we prove some properties of the argument structure $\mathcal{A}^*$ in the case that Σ is a Horn formula. In a first reading, the proofs can be skipped.

Lemma 1. *Let $\mathcal{A}$ be a formula in $\mathcal{C}_A$. A statement $h \in \mathcal{S}$ can be derived from $\mathcal{A} \wedge \Sigma$, i.e., $\mathcal{A} \wedge \Sigma \models h$, if and only if $h \in \mathcal{A}$ or if there is a Horn clause $s_1 \wedge \ldots \wedge s_m \rightarrow h$ in Σ such that $\mathcal{A} \wedge \Sigma \models s_i$, for $i = 1 \ldots m$.*

Proof. We first show that $f \models g$ if and only if $f \wedge \neg g$ is unsatisfiable. Consider any model $\mathcal{B}$ for f. If $f \models g$, then $\mathcal{B}$ is also a model for g. Hence $\hat{\mathcal{B}}(\neg g) = 0$ whenever $\hat{\mathcal{B}}(f) = 1$ and therefore $f \wedge \neg g$ is unsatisfiable. Conversely, that the formula $f \wedge \neg g$ is unsatisfiable means that if $\hat{\mathcal{B}}(f) = 1$ then $\hat{\mathcal{B}}(\neg g) = 0$. Hence whenever $\hat{\mathcal{B}}(f) = 1$ we have $\hat{\mathcal{B}}(g) = 1$. This corresponds to $f \models g$.

The problem of verifying $\mathcal{A} \wedge \Sigma \models h$ is therefore equivalent to deciding the unsatisfiability of $\mathcal{A} \wedge \Sigma \wedge \neg h$. The *marking algorithm* allows to determine the unsatisfiability of a formula $\mathcal{A} \wedge \Sigma \wedge \neg h$ [1]. Note that $\neg h$ is semantically equivalent to $h \rightarrow 0$, where 0 stands for an unsatisfiable formula.

Marking algorithm.

1. Rewrite $\mathcal{A} \wedge \Sigma \wedge \neg h$: Write all negative literals s_i in $\mathcal{A}$ as $s_i \rightarrow 0$ and $\neg h$ as $h \rightarrow 0$.
2. Mark all statements that occur positively in $\mathcal{A}$.
3. While there is a Horn clause $s_1 \wedge \ldots \wedge s_m \rightarrow B$ in Σ where the s_i for $i = 1 \ldots m$ are marked and B is not yet marked. If $B = 0$, return Unsatisfiable and stop. Else mark $B = s_{m+1}$.
4. Return Satisfiable and stop.

The time complexity of the algorithm is linear in the number of statements in $\mathcal{S}$: the while loop is executed at most $|\mathcal{S}|$ times. We show that the algorithm is correct: it returns Unsatisfiable if and only if $\mathcal{A} \wedge \Sigma \wedge \neg h$ is unsatisfiable. Thus h is logical consequence of $\mathcal{A} \wedge \Sigma$ if and only if $s \in \mathcal{A}$ or if there is a Horn clause $s_1 \wedge \ldots \wedge s_m \rightarrow h$ such that $\mathcal{A} \wedge \Sigma \models s_i$, for $i = 1 \ldots m$.

If the algorithm stops at step 3 and returns Unsatisfiable then the formula is unsatisfiable. For a model $\mathcal{B}$ of $\mathcal{A} \wedge \Sigma \wedge \neg h$ (if there is at all any model) one must have $\mathcal{B}(s) = 1$ for all statements s that have been marked during the algorithm: All statements that occur positively in $\mathcal{A}$ must be true and by definition of $\rightarrow$ also the postcondition if all of its precondiditions are true. The algorithm only returns Unsatisfiable if there is a Horn clause $s_1 \wedge \ldots s_m \rightarrow 0$. In this case, because the s_i are marked, for all possible models $\mathcal{B}$ of $\mathcal{A} \wedge \Sigma \wedge \neg h$, $\hat{\mathcal{B}}(s_1 \wedge \ldots \wedge s_n \rightarrow 0) = 0$, and therefore $\mathcal{A} \wedge \Sigma \wedge \neg h$ indeed is unsatisfiable.

On the other hand, if the algorithm stops at step 4 and returns Satisfiable, a model for $\mathcal{A} \wedge \Sigma \wedge \neg h$ exists, and hence $\mathcal{A} \wedge \Sigma \wedge \neg h$ is satisfiable. A model $\mathcal{B}$ is obtained by assigning the truth value 1 to all statements that have been marked, and 0 to the others. We have to show that $\hat{\mathcal{B}}(\mathcal{A} \wedge \Sigma \wedge (h \rightarrow 0)) = 1$.

Clearly, $\hat{\mathcal{B}}(\mathcal{A}) = 1$: all positive statements have been marked (at step 2); the negative statements have not been marked, because otherwise the algorithm would have stopped in 3. The same reasoning can be applied to h: $\mathcal{B}(h) = 0$ and therefore $\hat{\mathcal{B}}(\neg h) = 1$. We get $\hat{\mathcal{B}}(\Sigma) = 1$. Let f denote any Horn clause in Σ, $f = s_1 \wedge \ldots s_n \wedge \rightarrow s_{n+1}$. If s_{n+1} has not been marked, there is at least one s_j, j between 1 and n that has not been marked. Hence $\hat{\mathcal{B}}(f) = 1$ (and therefore $\hat{\mathcal{B}}(\Sigma) = 1$) since $\hat{\mathcal{B}}(s_j) = 0$ and $\hat{\mathcal{B}}(s_1 \wedge \ldots \wedge s_n) = 0$. ○

The proof of Lemma 1 shows that if $\mathcal{A} \wedge \Sigma \models h$, then this holds also for $Pos(\mathcal{A}) \wedge \Sigma \models h$, where $Pos(\mathcal{A})$ denotes the set of all positive literals occurring in $\mathcal{A}$. This implies that minimal arguments for h solely consist of positive literals.

Lemma 2. *Let $s_1 \wedge \ldots \wedge s_n \rightarrow s_{n+1}$ be a Horn clause in Σ. If $\mathcal{A}_i \wedge \Sigma \models s_i$, for $i = 1 \ldots n$, where the $\mathcal{A}_i$ consist only of positive literals, then $\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n$ is an argument for s_{n+1}: $\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n \wedge \Sigma \models s_{n+1}$.*

Proof. First observe that $\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n$ is satisfiable since the $\mathcal{A}_i$ solely consists of positive literals. $\mathcal{A}_i \wedge \Sigma \models s_i$ implies that $\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n \wedge \Sigma \models s_i$, for $i = 1 \ldots n$. This means that for all models $\mathcal{B}$ of $\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n \wedge \Sigma$ we have $\mathcal{B}(s_i) = 1$. In this case, $\hat{\mathcal{B}}(s_1 \wedge \ldots \wedge s_n \rightarrow s_{n+1}) = 1$ only if $\mathcal{B}(s_{n+1}) = 1$. Hence for all models of $s_1 \wedge \ldots \wedge s_n \rightarrow s_{n+1}$ we have $\mathcal{B}(s_{n+1}) = 1$ and therefore $\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n \wedge \Sigma \models s_{n+1}$. ○

Lemma 1 and 2 implicitly describe how an argument for h can be determined. Take a Horn clause where h occurs as postcondition. Determine an argument $\mathcal{A}_i$ for each of the preconditions s_i, $i = 1 \ldots n$. If s_i is an assumption, an argument for s_i is given by s_i itself; if not, recursively try to find an argument for s_i. If for all s_i there is an argument $\mathcal{A}_i$, $\mathcal{A} = \wedge_{i=1}^{n} \mathcal{A}_i$ is an argument for h.

Lemma 3. *If $h \in \mathcal{S} - \mathcal{E}$ does not occur as a precondition of a Horn clause in Σ, then there are no arguments for the hypothesis $\neg h$, i.e., there is no $\mathcal{A} \in \mathcal{C}_A$, such that $\mathcal{A} \wedge \Sigma \models \neg h$.*

Proof. We have to show that there is at least one model $\mathcal{B}$ for $\mathcal{A} \wedge \Sigma$ such that $\mathcal{B}(h) = 1$. By definition of an argument $\mathcal{A}$ for h, there is at least one model for $\mathcal{A} \wedge \Sigma$; if $\mathcal{B}(h) = 1$, we are done. Otherwise, consider the truth assignment $\mathcal{B}'$, such that $\mathcal{B}'(h) = 1$, and $\mathcal{B}'(s) = \mathcal{B}(s)$ for all statements s expect h. $\mathcal{B}'$ is also a model for $\mathcal{A} \wedge \Sigma$: $\hat{\mathcal{B}}'(\mathcal{A}) = 1$ (since h is not in $\mathcal{A}$, $\hat{\mathcal{B}}'(\mathcal{A}) = \hat{\mathcal{B}}(\mathcal{A})$). We have also that $\hat{\mathcal{B}}'(\Sigma) = 1$, since s occurs only as postcondition of a Horn clause. ○

Lemma 4. *If no assumption $a \in \mathcal{E}$ occurs as postcondition of a Horn clause in Σ, then there are no arguments for the contradiction, i.e., $\mathcal{A} \wedge \Sigma$ is always satisfiable.*

Proof. Let $\{s_1, \ldots, s_m\}$ be the set of all the m different statements occurring as postcondition of a Horn clause in Σ. By definition of an argument, $\mathcal{A}$ has at least one model $\mathcal{B}$. Consider the truth assignment $\mathcal{B}'$: $\mathcal{B}'(a_i) = \mathcal{B}(a_i)$ for all assumptions in $\mathcal{A}$, $\mathcal{B}'(s_i) = \mathcal{B}(s_i)$ for $i = 1 \ldots m$. Hence we have $\hat{\mathcal{B}}'(\mathcal{A}) = 1$ and $\hat{\mathcal{B}}'(\Sigma) = 1$. Since we can construct a model for $\mathcal{A} \wedge \Sigma$ for every $\mathcal{A} \in \mathcal{C}_A$ there is no argument for the contradiction. ○

2.4 Degrees of Uncertainty

Any uncertainty method uses a partially ordered set of values to represent degrees of uncertainty (or belief). For a given assumption a, one's belief can range from certainty that $\neg a$ is true over complete uncertainty to complete certainty that a is true. We assign confidence values only to non-negated assumptions, partial belief in a negated assumption can be represented by assigning a confidence value to a new non-negative assumption a' and introducing a new Horn clause $a' \rightarrow \neg a$.

Confidence values stand for the degree of certainty that a piece of evidence or a hypothesis is true. A *confidence set* $\mathcal{T}$ is a partially ordered set of confidence values, where the ordering is denoted by $<$. The ordering of the confidence values indicate the degree of certainty; a higher confidence value stands for more certainty. As usual, $t_1 \leq t_2$ stands for $t_1 < t_2$ or $t_1 = t_2$.

A confidence set contains a minimal and a maximal confidence value, denoted by $\bot$ and $\top$, respectively. The symbol $\bot$ stands for complete uncertainty ($\bot \leq t, \forall t \in \mathcal{T}$). The confidence value $\top$ stands for complete certainty ($t \leq \top, \forall t \in \mathcal{T}$) and captures an entity's belief that a statement is true.

A *confidence assignment* represents an entity's initial belief with respect to each of the assumptions. Formally, a confidence valuation is a function c from the set of pieces of evidence $\mathcal{E}$ to the set of confidence values $\mathcal{T}$:

$$c : \mathcal{E} \ \rightarrow \ \mathcal{T}$$

Let $\mathcal{C}$ denote the set of all confidence assignments. We assume that the confidence values assigned to the pieces of evidence are independent. Such an assumption is not restricting; in the approach described in this paper, dependencies between pieces of evidence could be encoded logically, i.e., as part of the evidence Σ. Consider the situation where the truth of the assumption a_1 depends on the truth of a_2 and vice versa. This dependency can be captured by introducing a statement (say a) and by replacing Σ by another formula Σ': $\Sigma' = \Sigma \wedge (a \wedge a_1 \rightarrow a_2) \wedge (a \wedge a_2 \rightarrow a_1)$. The degree of dependency between a_1 and a_2 is then captured by the confidence value $c(a)$ assigned to a.

A *confidence valuation* e is a function that takes as input a hypothesis (i.e, a statement in $\mathcal{S}$) and a confidence assignment and returns a confidence value for the hypothesis.

$$e : \mathcal{C} \times \mathcal{S} \ \rightarrow \ \mathcal{T}$$

3 Principles for Confidence Valuation

A confidence valuation reduces the *a priori* information (the confidence values assigned to the pieces of evidence) to a single confidence value for the hypothesis. The principles of the next section characterize the way a confidence valuation should combine the confidence values assigned to the pieces of evidence in order to obtain a confidence value for the hypothesis.

In the following, let $\mathcal{A}^*$ stand for the argument structure for h with respect to Σ. Our principles make sense if the argument structure $\mathcal{A}^*$ has the following two properties:

1. There is no argument for the counter-hypothesis $\neg h$.
2. Every argument $\mathcal{A}$ in $\mathcal{A}^*$ consists solely of positive literals.

We will explain why these properties for $\mathcal{A}^*$ are required when introducing our principles. Recall from Section 2 that if the hypothesis is a statement not belonging to the set of assumptions ($h \in \mathcal{S} - \mathcal{E}$) and if Σ is a Horn formula, then the argument structure has the above two properties.

If all arguments for h consist of at least one assumption which is completely uncertain for Alice, then Alice will be completely uncertain with respect to the truth of h. Conversely, if Alice is completely certain about the truth of all assumptions of one argument for h, then Alice will also be certain about the truth of h.

Principle 1. (Meaning of $\perp$ and $\top$.) If for all arguments $\mathcal{A}_i \in \mathcal{A}^*$ there is at least one assumption $a_{ij} \in \mathcal{A}_i$ such that $c(a_{ij})=\perp$, then

$$e(c, h) = \perp.$$

If there is at least one argument $\mathcal{A}_i \in \mathcal{A}^*$ such that for all assumptions a_{ij} in $\mathcal{A}_i$, $c(a_{ij})=\top$, then

$$e(c, h) = \top.$$

If one increases the confidence value for one piece of evidence then the confidence valuation should not return a lower confidence value for h:

Principle 2. (Monotonicity of e with respect to the confidence assignments.) Let c_1 and c_2 be two confidence assignments such that $c_1(a) \leq c_2(a)$ for all $a \in \mathcal{E}$. Then,

$$e(c_1, h) \leq e(c_2, h).$$

Note that this principle makes only sense if the arguments in $\mathcal{A}^*$ do not contain negative literals. If Alice's confidence in a increases, then Alice has less confidence in $\neg a$. If $\neg a$ is in an argument for h, then the hypothesis h is less supported and therefore h is less certain; therefore the result of the confidence valuation should decrease.

Two argument structures $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$ are called isomorphic if the assumptions in $\mathcal{A}_1^*$ can be renamed such that $\mathcal{A}_2^*$ is identical to $\mathcal{A}_1^*$ (in the sense of equality between sets). The notion of isomorphism captures our intuition in which case two argument structures for two hypotheses can be regarded to be equally strong: this is the case if $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$ are equal up to the names that have been chosen for the assumptions (or more generally for the statements).

Definition 1. *Let*

$$\mathcal{A}_1^* = \{\{s_{11}^1, \ldots, s_{1h}^1\}, \ldots, \{s_{m1}^1, \ldots, s_{mi}^1\}\}$$

and

$$\mathcal{A}_2^* = \{\{s_{11}^2, \ldots, s_{1j}^2\}, \ldots, \{s_{m1}^2, \ldots, s_{mk}^2\}\}$$

be two argument structures for two statements h_1 *and* h_2 *in* $\mathcal{S} - \mathcal{E}$*, respectively.* $\mathcal{A}_1^*$ *and* $\mathcal{A}_2^*$ *are* isomorphic *with respect to the function* $f \colon \mathcal{E} \rightarrow \mathcal{E}$ *if* f *is a bijection and*

$$\{\{f(s_{11}^1), \ldots, f(s_{1h}^1)\}, \ldots, \{f(s_{m1}^1), \ldots, f(s_{mi}^1)\}\} = \\ \{\{s_{11}^2, \ldots, s_{1j}^2\}, \ldots, \{s_{m1}^2, \ldots, s_{mk}^2\}\}.$$

Assume that there are two isomorphic argument structures $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$ for two hypotheses h_1 and h_2, and let f be the isomorphism. Clearly, if the argument structure for the two hypotheses are isomorphic and if Alice's confidence is equal for all assumptions that correspond to each other, then the result of the confidence valuation should in both cases be the same.

Principle 3. (Isomorphism of argument structures.) Let $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$ be two isomorphic argument structures for two hypotheses h_1 and h_2, respectively. Let f denote the corresponding bijection. If for all $a \in \mathcal{E}$,

$$c_1(a) = c_2(f(a))$$

then

$$e(c_1, h_1) = e(c_2, h_2).$$

Consider two arguments $\mathcal{A}_1$ and $\mathcal{A}_2$ for h where $\mathcal{A}_1 \subset \mathcal{A}_2$. (Recall that arguments are represented as sets). Intuitively, $\mathcal{A}_1$ is a stronger argument than $\mathcal{A}_2$ since in the case of $\mathcal{A}_1$, less assumptions must be true such that one can derive h. Formally, $\mathcal{A}_1 \subset \mathcal{A}_2$ implies that if $\mathcal{A}_1 \wedge \Sigma \models h$ then also $\mathcal{A}_2 \wedge \Sigma \models h$.

The next principle states that if for all arguments $\mathcal{A}_1$ in $\mathcal{A}_1^*$ we can find a stronger argument $\mathcal{A}_2$ in $\mathcal{A}_2^*$, then the output of the confidence valuation should be equal or higher in the latter case.

Principle 4. (Implication.) Let $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$ be the two argument structures for two hypotheses h_1 and h_2, respectively. If for all argument $\mathcal{A}_1 \in \mathcal{A}_1^*$ there is an argument $\mathcal{A}_2 \in \mathcal{A}_2^*$ such that $\mathcal{A}_1 \supseteq \mathcal{A}_2$ then, for any confidence assignment c,

$$e(c, h_1) \leq \ e(c, h_2).$$

4 Valuating Public-Key Authenticity

4.1 Modeling Public-Key Certification

In our approach, modeling a certain problem consists of identifying the pieces of evidence (i.e., $\mathcal{E}$), the possible conclusions (i.e., $\mathcal{S}$), and describing how the truth values of the evidence and the conclusion is related (i.e., Σ).

Different authentication methods are based on different models. The model of Reiter and Stubblebine consists of a set of public-key certificates [16]. PGP's method takes trust values with respect to keys into account, and in Maurer's model one assigns confidence values for the trustworthiness of a person [11] (see also below).

Since authenticating public keys is uncertain also because entities in a web of trust can misbehave (by issuing "wrong" certificates) an authentication method must rely on trust values that one assigns to persons. The problem that arises if trust is not with respect to persons but with respect to keys is best illustrated by the simple example depicted in Figure 2. In the left scenario Bob's key is accepted to be `valid` while in the right scenario it is not. Even if Carol is only `marginally trusted`, she can make Alice accept Bob's key to be `valid` by issuing certificates with two different keys. This is against the intention of the designer of PGP's method that two introducers are needed if they are only `marginally trusted`.

In the left scenario the assertions made by means of the public-key certificates and signed with K_1 and K_2 are treated as if they were independent whereas clearly they are not. K_1 and K_2 are both controlled by Carol and hence both certificates for Bob's key have been issued by Carol. If trust values would be assigned to persons, both certification path (and hence both arguments) would depend on the trust value assigned to Carol.

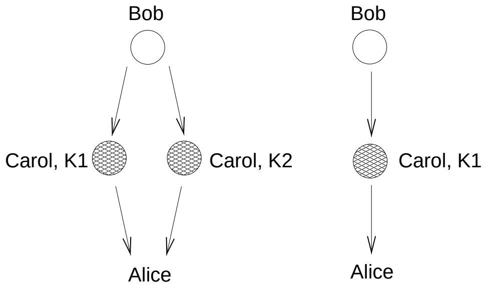

Fig. 2. A `marginally trusted` entity using multiple keys.

The same remark holds for the confidence valuations proposed by Reiter and Stubblebine [16]. A single person can produce any result for the confidence valuation, by generating an arbitrary number of keys and then by issuing public-key certificates.

4.2 Confidence Valuation in PGP 2.6.2

We first formalize PGP's authentication method. The evidence and the conclusions consist of three types of statements, which we denote by $Aut_{X,K}$, $Cert_{K_1,X,K_2}$ and $Trust_{X,K}$. $Aut_{X,K}$ stands for the fact that the signature public key K is authentic for the entity X. $Cert_{K_1,X,K_2}$ means there exists a public-key certificate, signed with the signature key K_1, claiming that K_2 is authentic for X. Finally, $Trust_{X,K}$ stands for the fact that X is trustworthy to provide authentic public keys of other entities by means of a public key K. A key K is authentic for a person X if there is a certificate issued by Y by means of a public key K_1, and if Y is trustworthy to issue public-key certificates. Therefore Σ consists of Horn clauses of the following form:

$$Aut_{Y,K_1} \wedge Trust_{Y,K_1} \wedge Cert_{K_1,X,K} \rightarrow Aut_{X,K}$$

The confidence valuation has been informally described in Section 1 and is formalized in Appendix A.

PGP's confidence valuation follows principle 1. If on every path there is an entity whose trust value is `no trust` or if there is a key that is `not valid` then the target key is `not valid`. Conversely, if there is a path where all entities are `fully trusted` and where all keys are `valid` then the key is accepted to be `valid`.

Principle 2 is also satisfied by PGP's confidence valuation. If PGP evaluates a key to be `valid`, it will still be valid when Alice increases a confidence value for a statement in $\mathcal{E}$, that is if Alice modifies her public-key ring in the following way: Alice increases a trust value, signs a public key with her own key, or adds a public-key certificate.

Principle 3 is not met by PGP. Again, consider the two scenarios of Figure 1 and assume that the entity X_i "presumably" controls the key K_i. Here, the evidence consists of the following statements:

$$\begin{aligned}\mathcal{E} = \{\ & Aut_{X_1,K_1}, Aut_{X_2,K_2}, Cert_{K_1,X_3,K_3}, Cert_{K_2,X_4,K_4}, Cert_{K_3,B,K_B}, Cert_{X_4,B,K_B} \\ & Trust_{X_1,K_1}, Trust_{X_2,K_2}, Trust_{X_3,K_3}, Trust_{X_4,K_4}\}.\end{aligned}$$

Σ is obtained by instantiating the above Horn formula with the statements in $\mathcal{E}$:

$$\begin{aligned}\Sigma = & (Aut_{X_1,K_1} \wedge Cert_{K_1,X_3,K_3} \wedge Trust_{X_1,K_1} \rightarrow Aut_{X_3,K_3}) \wedge \\ & (Aut_{X_2,K_2} \wedge Cert_{K_2,X_4,K_4} \wedge Trust_{X_2,K_2} \rightarrow Aut_{X_4,K_4}) \wedge \\ & (Aut_{X_3,K_3} \wedge Cert_{K_3,B,K_B} \wedge Trust_{X_3,K_3} \rightarrow Aut_{B,K_B}) \wedge \\ & (Aut_{X_4,K_4} \wedge Cert_{K_4,B,K_B} \wedge Trust_{X_4,K_4} \rightarrow Aut_{B,K_B}).\end{aligned}$$

The confidence assignment can be read out from the web of trust. For instance, in the left scenario, we have the following confidence assignment c_l (where `ft` stands for `fully trusted` and `mt` for `marginally trusted`):

Aut_{X_1,K_1}	Aut_{X_1,K_1}	$Trust_{X_1,K_1}$	$Trust_{X_2,K_2}$	$Trust_{X_3,K_3}$
`val`	`val`	`ft`	`ft`	`mt`
$Trust_{X_4,K_4}$	$Cert_{K_1,X_3,K_3}$	$Cert_{K_2,X_4,K_4}$	$Cert_{K_3,B,K_B}$	$Cert_{K_4,B,K_B}$
`mt`	`val`	`val`	`val`	`val`

An argument for $Aut_{X,K}$ simply corresponds to the set of statements that are on the paths from the source key (in our examples Alice) to the target key (which is allegedly controlled by X). Both scenarios of Figure 1 have the same argument structure for Aut_{B,K_B}, and the two argument structures are therefore isomorphic. The two arguments for Aut_{B,K_B} are

$$\begin{aligned}\mathcal{A}_1 &= \{Aut_{X_1,K_1}, Cert_{K_1,X_3,K_3}, Cert_{K_3,B,K_B}, Trust_{X_1,K_1}, Trust_{X_3,K_3}\},\\ \mathcal{A}_2 &= \{Aut_{X_2,K_2}, Cert_{K_2,X_4,K_4}, Cert_{X_4,B,K_B}, Trust_{X_2,K_2}, Trust_{X_4,K_4}\}.\end{aligned}$$

There is a bijection $f : \mathcal{E} \to \mathcal{E}$:

$Trust_{X_1,K_1}$	$Trust_{X_2,K_2}$	$Trust_{X_3,K_3}$	$Trust_{X_4,K_4}$	$f(x)$
$Trust_{X_2,K_2}$	$Trust_{X_1,K_1}$	$Trust_{X_4,K_4}$	$Trust_{X_3,K_3}$	x else.

such that $c_l(a) = c_r(f(a))$. Thus, according to Principle 3, one could postulate that the confidence valuation should return the same confidence value for both scenarios. However, this is not the case, as mentioned in Section 1.

Assume that for every argument $\mathcal{A}_1$ of a hypothesis Aut_{X_1,K_1} there is an argument $\mathcal{A}_2$ for Aut_{X_2,K_2} such that $\mathcal{A}_1 \supseteq \mathcal{A}_2$. This means that for every path $\mathcal{A}_1$ of K_1 there is a path $\mathcal{A}_2$ for K_2 such that $\mathcal{A}_2$ is a sub-path of $\mathcal{A}_1$. In this case, the key K_1 is not accepted to be `valid` if the key K_2 is not accepted to be `valid`. Hence principle 4 is followed by PGP.

4.3 Maurer's Confidence Valuation

Maurer's model of a public-key infrastructure consists of two parts [11]. In the deterministic part, he identifies the pieces of evidence that a user allow to derive that a public key is authentic for a certain entity. The deterministic part corresponds to what we here call the model of the confidence valuation (see Subsection 4.1). In the probabilistic part, probabilities stand for degrees of uncertainty. His method is based on a well-defined random experiment. The probabilistic method is inspired from other uncertainty methods, where probabilities stand for degrees of uncertainty or subjective belief [18, 5].

Maurer uses a similar set of statements as PGP. In his model, however, trust is with respect to persons and not with respect to keys. Secondly, recommendations are part of the model: by means of a digitally signed statement, introducers can not only assert that a certain public key is authentic for a certain entity, but they can also recommend other entities to be trustworthy. For simplicity, we will describe a version of his model where we do not consider recommendations. The observations that we will make are nevertheless valid for the complete model.

The model consists of the following types of statements [11]. $Aut_{A,B}$ stands for A's belief that she holds an authentic public key of B. $Trust_{A,B}$ means that A believes that B is trustworthy. $Cert_{X,Y}$ stands for the fact that X has issued a public-key certificate for Y.

The fact that only entities and not keys are parameters in Maurer's statements can raise confusion and has been criticized by Reiter and Stubblebine [16]. As they state, "entities don't sign certificates, keys do" (principle 1 in [16]). In Maurer's model there is the implicit assumption that every entity controls only one key; therefore it is not explicitly mentioned which key is concerned. For instance, the statements $Cert_{X_1,Y}$ and $Cert_{X_2,Y}$ stand for the fact that X_1 and X_2 have issued a certificate for the same key of Y. This confusion could be avoided by explicitly introducing statements where the keys that are concerned are mentioned, as it is the case in our formalization of PGP. One would also obtain a more realistic model where an entity can hold more than one key.

$View_A$ is the set of pieces of evidence that A collected. From $View_A$, Alice tries to derive statements by recursively applying the following inference rule:

$$Aut_{A,X} \wedge Trust_{A,X} \wedge Cert_{X,B} \vdash Aut_{A,B}$$

The statement $Aut_{A,B}$ can be derived if $Aut_{A,X}$, $Trust_{A,X}$ and $Cert_{X,B}$ are in $View_A$ or, recursively, if they are derivable by applying the inference rule. $\overline{View_A}$ stands for the set of statements that are initially in $View_A$ or that can be derived from $View_A$.

Note that this derivation procedure is purely syntactic, and that Maurer therefore uses the symbol $\vdash$ rather than $\rightarrow$ to denote that a statement follows from a set of other statements. The notion of derivability in Maurer's model corresponds to our notion of logical consequence, in the following way. $View_A$ corresponds to $\mathcal{E}$ and $\overline{View_A}$ to $\mathcal{S}$. Σ is obtained by instantiating the above inference rule with the statements in $\mathcal{S}$. As one can show, the statement s is derivable from $View_A$ (i.e., $s \in \overline{View_A}$) if and only if $\mathcal{E} \wedge \Sigma \models s$. A minimal set of statements $\mathcal{V}$ such that the statement s can be derived from $\mathcal{V}$ is called a path. Obviously, the notion of a path corresponds to our notion of a minimal argument, and the set of paths of $Aut_{A,B}$ corresponds to the argument structure.

In the sequel, let $\mathcal{S}_A$ stand for the set of statements that are in Alice's view. In the probabilistic part, $View_A$ is interpreted as a random variable where the domain is the powerset of $\mathcal{S}_A$, i.e., $View_A$ takes as values subsets of $\mathcal{S}_A$. Alice expresses her uncertainty towards the pieces of evidence by specifying a probability distribution for $View_A$. In order to keep the number of confidence values that Alice must assign reasonably small, one can make the assumption that the pieces of evidence are independent. This means in particular that the introducers are assumed not to collude. In case of the independence assumption, Alice assigns a probability to each piece of evidence.

Since $View_A$ is a random variable, also $\overline{View_A}$ is a random variable. The confidence value for a hypothesis h is defined as the probability that h can be derived from $View_A$:

$$e(c,h) = P(h \in \overline{View_A}).$$

To compute $e(c, h)$, one can determine all minimal arguments for h. Let $\mathcal{V}_i$, $i = 1 \ldots k$, stand for the k minimal arguments. $\mathcal{V}_i \subseteq \mathit{View}_A$ stands for the event that h can be derived from $\mathcal{V}_i$. The confidence value $e(c, h)$ is obtained by computing the probability that h can be derived from at least one argument, i.e., by computing the probability of the the union of the events $\mathcal{V}_i \subseteq \mathit{View}_A$, i.e.,

$$e(c, h) = P(\bigvee_{i=1}^{k} (\mathcal{V}_i \subseteq \mathit{View}_A)).$$

The probability $P(\mathcal{V}_i \subseteq \mathit{View}_A)$ is the product of the probabilities of the statements in $\mathcal{V}_i$, in case these probabilities are independent. Since the events $\mathcal{V}_i \subseteq \mathit{View}_A$ intersect, one cannot simply add up the probabilities $P(\mathcal{V}_i \subseteq \mathit{View}_A)$. A naive approach would be to establish a table where each row is indexed by a subset of $\mathcal{S}_A$. For each row $\mathcal{V} \subseteq \mathcal{S}_A$, one would store

$$P(\mathit{View}_A = \mathcal{V}) = \prod_{s \in \mathcal{V}} p(s) \prod_{s \notin \mathcal{V}} (1 - p(s)).$$

e(c,h) is the sum of the probabilities of all $\mathcal{V}$ from which s can be derived:

$$e(c, h) = \sum_{s \in \bar{\mathcal{V}}} P(\mathit{View}_A = \mathcal{V}).$$

More efficiently, the union of the events $P(\mathcal{V}_i \subseteq \mathit{View}_A)$ can be computed according to the inclusion-exclusion principle (see [11]).

Maurer's confidence valuation satisfies our principles.

Principle 1. By assumption for every argument $\mathcal{V}_i$ there is a statement a such that $c(a) = 0$. Hence $P(\mathcal{V}_i \subseteq \mathit{View}_A) = 0$, for $i = 1 \ldots k$. The probability of the union of events is 0 if the probability of all events is 0, and therefore $e(c, s) = P(\vee_{i=1}^{k} (\mathcal{V}_i \subseteq \mathit{View}_A)) = 0$.

Conversely, assume that there is an argument $\mathcal{V}_i$ such that $c(a) = 1$ for all statements a in $\mathcal{V}_i$. Hence $P(\mathcal{V}_i \subseteq \mathit{View}_A) = 1$. Since the probability of the union of events is always bigger than the probability of one event we get $e(c, s) = 1$.

Principle 2. By increasing the probability of one statement a (i.e, $c'(a) > c(a)$) one gets $e(c, h) \geq e(c', h)$. Let $\mathcal{V}_{-a}$ denote the set $\mathcal{V} - a$ and $\mathcal{V}_{+a}$ denote the set $\mathcal{V} \cup a$. Observe that if h can be derived from $\mathcal{V}_{-a}$, then it can also be derived from $\mathcal{V}_{+a}$ (but the inverse is not necessarily true). Therefore one can distinguish two cases. If h can be derived from $\mathcal{V}_{-a}$ then $P(\mathit{View}_A = \mathcal{V}_{-a}) + P(\mathit{View}_A = \mathcal{V}_{+a})$ does not depend on the value $c(a)$. But there might exist rows such that h can not be derived from $\mathcal{V}_{-a}$ but from $\mathcal{V}_{+a}$; moreover the higher $c(a)$, the higher $P(\mathit{View}_A = \mathcal{V}_{+a})$.

Principle 3. Assume that the argument structures $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$ for two hypotheses Aut_{A,B_1} and Aut_{A,B_2} are isomorphic with respect to f. Additionally, assume that there is twice the same probability distribution (i.e, $c_1(a) = c_2(f(a))$ for all $a \in \mathcal{E}$). Since the argument structures are isomorphic, one has to add up the same number of rows in order to compute the confidence value for Aut_{A,B_1} as for Aut_{A,B_2}. Moreover, since there is twice the same probability distribution, Maurer's confidence valuation will twice return the same result. Therefore it satisfies principle 3.

Principle 4. Assume that for every minimal argument $\mathcal{V}_1$ of Aut_{A,B_1} there is a minimal argument $\mathcal{V}_2$ of Aut_{A,B_2} such that $\mathcal{V}_1 \supseteq \mathcal{V}_2$. This implies that for computing the confidence value of Aut_{A,B_2} one has to add up more rows than for Aut_{A,B_1}. Hence principle 4 is satisfied.

5 Conclusions

5.1 The Deficiency of Extensional Methods

One possible criterion to classify uncertainty methods is whether the uncertainty is dealt with *extensionally* or *intensionally* [13]. In extensional systems, the uncertainty of a formula is computed as a function of the uncertainty of its subformulas. In other words, the confidence value of a conclusion is a function of the confidence values of the preconditions of the rule [13]. In intensional systems, uncertainty is attached to "state of affairs" or "possible worlds". There seems to be a trade-off between computational efficiency and semantic correctness: Extensional systems have the advantage of generally being computationally more efficient than intensional systems. On the other hand, extensional systems often suffer the deficiency to produce counter-intuitive conclusions [13].

PGP is a representative of an extensional system since the validity of a key is computed as a function of the trust values attached to the signature keys under the public key. From this perspective, it is not surprising that one can construct scenarios where PGP returns counter-intuitive results. Maurer's approach is an example of an intensional system, because Alice specifies a probability distribution over her possible views, and the confidence value for the hypothesis is the probability that the hypothesis can be derived from the view.

5.2 The Principles of Reiter-Stubblebine

Reiter and Stubblebine also introduce principles for a method of authentication (the RS principles for short) [16], and it is appropriate to compare their work with ours. Their principles can be understood as general guidelines summarizing common sense, at the price of being vague, while our principles are formulated within a precise mathematical framework, hence more precise, at the price of being less comprehensive.

For instance, what it means for "the output of a metric to be intuitive" (in RS-principle 4) is made precise in this paper by the principles for the confidence valuation. As a second example, the RS-principle 5 ("A metric should be resilient to manipulation of its model by misbehaving entities, and its sensitivity to various forms of misbehavior should be made explicit.") is also vague because it is not made precise what the model of a metric is and in what ways the model can be manipulated. We characterize the reliability of a web of trust by the argument structure for the given evidence and hypothesis. The RS-principles 1, 3, 5 and 6 concern the modeling of evidence rather than how to deal with uncertainty. RS-principle 7 ("A metric should be able to be computed efficiently.") is again very general but of course everybody would agree to it.

5.3 A Direction for Future Research

Our principles are a first natural characterization of how a confidence valuation should combine initial confidence values. They allow to point out problems arising in PGP's trust management. It is not our claim that we provide a complete characterization of a confidence valuation. Figure 3 shows a pair of scenarios where PGP's confidence valuation apparently produces a counter-intuitive result, even if it does not violate any of our current principles. In the left scenario Bob's key is considered to be `invalid`, while in the right scenario it is considered to be `valid`. However, the left scenario seems to be more secure than the right one. In the left scenario X_3 and X_4 have to collude in order to palm off a wrong key for Bob on Alice, whereas in right scenario X_3 can achieve this alone. A research goal is to find a complete characterization of a confidence valuation; this is not only of interest in the context of applied cryptography, but more generally in artificial intelligence and evidence theory.

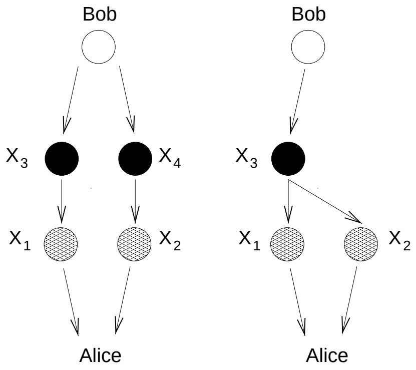

Fig. 3. Another inconsistence in PGP.

Acknowledgments

We wish to thank Christian Cachin, Rolf Haenni and Bartosz Przydatek for interesting discussions and helpful comments.

References

1. Nerode A. and Shore R. A.˙ *Logic for applications.* Springer-Verlag, 1993.
2. J. Bernoulli. Ars conjectandi, 1713. Reprinted in 1968 by Culture et Civilisation, 115 Avenue Gabriel Lebon, Brussels.
3. T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open systems. In D. Gollmann, editor, *Computer Security - ESORICS'94*, volume 875 of *Lecture Notes in Computer Science*, pages 3–18. Springer Verlag, Berlin, 1994.
4. J. de Kleer. An assumption-based TMS. *Artificial Intelligence, Elsevier Science Publisher B.V. (Amsterdam)*, 28:127–162, 1986.
5. R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems, 1999.
6. M. Henrion, H. J. Suermondt, and D. E. Herckermann. Probabilistic and Bayesian representations of uncertainty in information systems: A pragmatic introduction. In Amihai Motro and Phillipe Smets, editors, *Uncertainty management in information systems*, chapter 9. Kluwer Academic Press, 1997.
7. J. Kohlas and P.-A. Monney. A mathematical theory of hints. In *Lecture Notes in Economics and Mathematical Systems.*, volume 425. Springer, 1995.
8. R. Kohlas and U. M. Maurer. Reasoning about public-key certification - on bindings between entities and public keys. In M. Franklin, editor, *Financial Cryptography 99*, LNCS, 1999.
9. R. Kruse, E.Schwecke, and J.Heinsohn. *Uncertainty and Vagueness in Knowlege Based Systems.* Springer Verlag, 1991.
10. E. H. Mamdani. On the classification of uncertainty techniques in relation to the application needs. In Amihai Motro and Phillipe Smets, editors, *Uncertainty management in information systems*, chapter 14. Kluwer Academic Press, 1997.
11. U. M. Maurer. Modelling a public-key infrastructure. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *Proceedings 1996 European Symposium on Research in Computer Security (ESORICS' 96), Lecture Notes in Computer Science, Springer*, LNCS, pages 325–350, 1996.
12. N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–86, 1986.
13. J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers, Inc., 1988.
14. M. K. Reiter and S. G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2), MAY 1997.
15. M. K. Reiter and S. G. Stubblebine. Path independence for authentication in large-scale systems. *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 57–66, 1997.
16. M. K. Reiter and S. G. Stubblebine. Toward acceptable metrics of authentication. *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 10–20, 1997.
17. A. Jøsang. An algebra for assessing trust in certification chains. In *Network and Distributed Systems Security (NDSS'99)*, 1999.
18. G. Shafer. Non-additive probabilities in the work of Bernoulli and Lambert.
19. G. Shafer. *A mathematical Theory of Evidence.* Princeton University Press, 1996.
20. W. Stallings. *Protect your privacy.* Prentice Hall, 1996.
21. A. Tarah and C. Huitema. Associating metrics to certification paths. In *Computer Security ESORICS 92*, Lecture Notes in Computer Science, pages 175–189. Springer Verlag, Berlin, 1992.
22. P. R. Zimmermann. *The Official PGP User's Guide.* MIT Press, Cambridge, MA, USA, 1995.

A PGP's Confidence Valuation (Version 2.6.2)

Formally, PGP's confidence valuation can be described as follows:

$$e(c, Aut_{X,K}) = \begin{cases} \texttt{valid} & c(Aut_{X,K}) = \texttt{valid} \quad \text{or} \\ & \exists K_1, Y | c(Cert_{K_1,X,K}) = \texttt{valid} \wedge \\ & c(Trust_{Y,K_1}) = \texttt{ultimately trusted} \quad \text{or} \\ & \exists K_1, Y | c(Cert_{K_1,X,K}) = \texttt{valid} \wedge \\ & c(Trust_{Y,K_1}) = \texttt{fully trusted} \wedge \\ & e(c, Aut_{Y,K_1}) = \texttt{valid} \quad \text{or} \\ & \exists K_1, K_2, Y, Z | c(Cert_{K_1,X,K}) = \texttt{valid} \wedge \\ & c(Cert_{K_2,X,K}) = \texttt{valid} \wedge \\ & c(Trust_{Y,K_1}) = \texttt{marginally trusted} \wedge \\ & c(Trust_{Z,K_2}) = \texttt{marginally trusted} \wedge \\ & e(c, Aut_{Y,K_1}) = \texttt{valid} \wedge \\ & e(c, Aut_{Z,K_2}) = \texttt{valid} \\ \texttt{not valid} & \text{else.} \end{cases}$$