

# Credential Networks: a General Model for Distributed Trust and Authenticity Management

Jacek Jonczy and Rolf Haenni  
University of Berne  
Institute of Computer Science and Applied Mathematics  
CH-3012 Berne, Switzerland  
{jonczy,haenni}@iam.unibe.ch

## Abstract

*In large open networks, handling trust and authenticity adequately is an important prerequisite for security. In a distributed approach, all network users are allowed to issue various types of credentials, e.g. certificates, recommendations, revocations, ratings, etc. This paper proposes such a distributed approach, in which the evaluation of trust and authenticity is based on so-called credential networks. The corresponding formal model includes many existing trust models as special cases.*<sup>1</sup>

## 1. Introduction

In the modern information society, large open and distributed networks become more and more important. The internet itself is such a network, and many internet applications such as electronic mailing, peer-to-peer file sharing, internet phoning, online auctions, online games, etc. form corresponding sub-networks. From the perspective of a single user, a common security problem of such networks is the fact that most or all other users of the network are unknown. Communicating with an unknown user  $X$  brings up (at least) two crucial questions:

- (1) What is the real identity of  $X$ ? Is he/she the one he/she claims to be?
- (2) How reliable is  $X$ ? Is it secure to use the service he/she offers?

The first question concerns the *authenticity* of the available information about  $X$ 's identity, whereas the second question is about whether or not  $X$  is a *trustworthy* service provider. In other words, (1) is about what  $X$  is, whereas (2) is about what  $X$  does.

<sup>1</sup>Research supported by the Swiss National Science Foundation, Project No. PP002-102652/1.

Properly managing authenticity and trust in a distributed network is anything but trivial. Most approaches are based on a corresponding formal model. There is a general distinction between *centralized* and *decentralized* models. In the former case, the responsibility of issuing various forms of *credentials*<sup>2</sup> is taken over by a central authority. One can think of a credential as a digitally signed statement or attestation about what another user is or does. A centralized model usually requires all network users to fully trust the central authority, whereas in a decentralized model, every user is also a potential issuer of such credentials. A given set of credentials, possibly issued by many different users, forms a so-called *credential network*. Note that centralized models can be regarded as special cases of decentralized ones.

This paper proposes credential networks as a new model for distributed authenticity and trust management. It is based on six different basic types of credentials. The *weight* of a credential is a numerical value assigned by the issuer to express various levels of confidence. The goal is to use credential networks for a qualitative and quantitative evaluation of trust and authenticity. As we will see, many existing models are included as special cases. For the underlying mathematical framework, we propose to use the theory of *probabilistic argumentation* [21, 23], which is a unified theory of logical and probabilistic reasoning. A short introduction to probabilistic argumentation is given in Subsection 1.2.

### 1.1. Related Work

The origin of the approach proposed in this paper is the so-called *web of trust* model introduced by PGP, a popular application for email security [43]. PGP organizes public keys

<sup>2</sup>*Credentials* are the central formal concepts of this paper. It is a generic term used for certificates, recommendations, revocations, discredits, and ratings. Precise definitions will be given in Section 2.

and corresponding certificates in local *key rings*. The owner of the key ring obtains a web of trust by assigning trust values to all certificate issuers. This forms then the basis for a qualitative evaluation of the authenticity of the public keys involved. In PGP, the evaluation of a web of trust is based on three simple rules and produces two possible outcomes, namely that a key is *valid* or *invalid*. It has been shown that these constraints are unsatisfactory [20]. Another problem is the restriction to only three trust values *completely trustworthy*, *marginally trustworthy*, and *untrustworthy*.

The model proposed by Haenni [20, 22] is similar to the PGP web of trust, but with a continuous range of possible trust values between 0 (no trust) and 1 (full trust), it offers a better flexibility. These values are interpreted as probabilities (of reliability), i.e. the evaluation turns out to be a problem of probabilistic reasoning. This model also proposes a general form of *key revocations*, which are understood as negative certificates. As a consequence, it may happen that the key ring contains certificates and revocations for the same public key, which requires a proper conflict management. A possible extension of Haenni’s model is the inclusion of *recommendations*, possibly on different levels, but the discussion in [22] is preliminary. Recommendations will therefore not be considered to be part of Haenni’s model.

Another existing model is the one proposed by Maurer [30, 32]. With respect to Haenni’s model, it is more general in two ways. First of all, Maurer’s model includes both certificates and recommendations, the latter on different levels. A recommendation of level  $i$  certifies its recipient to be trustworthy in giving recommendations of level  $i - 1$ . In this sense, certificates are interpreted as particular recommendations of level 0. The second more general feature of Maurer’s model is the possibility to attach a *confidence parameter* whenever a certificate or recommendation is issued. In PGP and in Haenni’s model, certificates are always absolute, which corresponds to the particular case in which all confidence parameters are set to 1. Note that Maurer’s model does not support negative certificates (revocations) or negative recommendations (discredits).

Numerous other trust models and trust metrics have been proposed [5, 6, 7, 8, 9, 10, 14, 25, 26, 35, 40]. Many of them focus on recommendations and do not require them to be digitally signed by the issuer. This is a problematical simplification and opens the door for forgeries and frauds, but depending on the security requirements of the intended application, it may be acceptable. The resulting *reputation networks* are widely discussed in the areas of electronic commerce [15, 27, 29, 31, 33, 41, 42] and peer-to-peer networks [1, 2, 17, 28, 38, 39].

## 1.2. Probabilistic Argumentation

*Probabilistic argumentation* is a relatively new theory of automated reasoning under uncertainty, in which the concepts of *provability* in logic and *probability* in probability theory are replaced by a more general concept of *probability of provability*. This is a sub-additive measure of how much a hypothesis or its complement is *supported* by the given knowledge. Accordingly, we will refer to it as the *degree of support* of a hypothesis. Formally, one can also think of probabilistic weights of sets of arguments (or counter-arguments) which, if assumed to be true, logically entail the hypothesis. It can be shown that this approach unifies the classical fields of logical and probabilistic reasoning [21].

The ingredients of probabilistic argumentation are a formal language  $\mathcal{L}_V$  over a set of variables  $V$  and a fully specified probability distribution  $\mathbf{P}(W)$  over a subset of variables  $W \subseteq V$ . The elements of  $W$  are called *probabilistic variables*. In this paper, we will only consider the simplest case of a possible formal language, namely the language of *propositional logic*. We will therefore consider  $V$  and  $W$  to be sets of propositions rather than sets of (Boolean) variables. Furthermore, we suppose the available information to be encoded by a set  $\Sigma \subseteq \mathcal{L}_V$  of propositional sentences (clauses). A tuple  $\mathcal{S} = (V, W, \mathbf{P}, \Sigma)$  is called *probabilistic argumentation system* (PAS). Note that probabilistic argumentation degenerates into logical reasoning for  $W = \emptyset$  and into probabilistic reasoning for  $W = V$ .

With respect to a hypothesis  $h \in \mathcal{L}_V$ , the goal of probabilistic argumentation is twofold. From a *qualitative* point of view, it consists of finding respective *arguments* and *counter-arguments* for  $h$ . Formally, if  $L_W = W \cup \{\neg x : x \in W\}$  denotes the set of all literals of  $W$ , then an argument is a set  $\alpha \subseteq L_W$  of literals for which

$$\alpha \cup \Sigma \models h \quad (1)$$

holds, and in which every proposition appears at most once. One can think of an argument as a collection of assumptions that make the hypothesis true in the light of  $\Sigma$ . In this sense, every argument *supports* and every counter-argument *weakens* the hypothesis. Note that counter-arguments are arguments for  $\neg h$ . The sets of all arguments and counter-arguments for  $h$  are denoted by  $Args(h)$  and  $Args(\neg h)$ , respectively.

If  $\alpha$  is both argument and counter-argument for  $h$ , then it is called *conflict*. Conflicts are inconsistent with respect to the given information  $\Sigma$ , i.e. they reflect impossible states of the world which have to be excluded. Conflicts can be regarded as arguments for  $\perp$ .

An argument or conflict  $\alpha$  for  $h$  is called *minimal*, if there is no shorter argument  $\alpha' \subset \alpha$  for  $h$ . If  $T$  is an arbitrary set of arguments, then  $\mu T = \{\alpha \in T : \nexists \alpha' \subset \alpha, \alpha' \in T\}$  denotes the corresponding *minimal*

set. With respect to  $h$ , the minimal sets of arguments and counter-arguments are denoted by  $args(h) = \mu Args(h)$  and  $args(\neg h) = \mu Args(\neg h)$ , respectively. Similarly,  $args(\perp) = \mu Args(\perp)$  stands for the the minimal set of conflicts.

Computing the sets  $args(h)$ ,  $args(\neg h)$ , and  $args(\perp)$  is the main computational problem of probabilistic argumentation [23]. Efficient approximation algorithms are obtained by focussing the search on the most relevant arguments [18]. It is also possible to define convenient *anytime algorithms* which, upon interruption, return an approximate solution [19]. In Subsection 4.3, we will present a simple special purpose algorithm designed for the specific problem addressed in this paper.

From a *quantitative* point of view, the goal of probabilistic argumentation is to measure the strength of a given set of arguments on the basis of the probability distribution  $\mathbf{P}(W)$ . The idea is to consider the conditional probability that at least one argument for  $h$  is true under the condition that none of the conflicts is true. If we think of the sets  $args(h)$  and  $args(\perp)$  as DNFs (disjunctive normal forms), then  $args(h)$  is a logical representation of the event that at least one argument is true, whereas  $\neg args(\perp)$  represents the condition that conflicts are impossible. This allows us to define *degree of support* of  $h$  as

$$\begin{aligned} dsp(h) &= P(args(h) \mid \neg args(\perp)) \\ &= \frac{P(args(h)) - P(args(\perp))}{1 - P(args(\perp))}. \end{aligned} \quad (2)$$

For a more detailed derivation of the above formula we refer to [23]. Degrees of support are *probabilities of provability* [34, 37], i.e. ordinary probabilities of a particular kind of events, namely that a hypothesis is provable in the light of  $\Sigma$ . It forms a *non-monotone* and *non-additive* measure of uncertainty and ignorance [21].

A second way of judging the hypothesis  $h$  quantitatively is to look at the conditional probability that no counter-argument is true under the condition that none of the conflicts of  $\Sigma$  is true. This is a quantitative measure of how possible  $h$  is. Accordingly, we define *degree of possibility* of  $h$  as

$$\begin{aligned} dps(h) &= P(\neg args(\neg h) \mid \neg args(\perp)) \\ &= \frac{1 - P(args(\neg h))}{1 - P(args(\perp))} = 1 - dsp(\neg h). \end{aligned} \quad (3)$$

This definition implies  $dsp(h) \leq dps(h)$  for all  $h \in \mathcal{L}_V$  and  $\Sigma \subseteq \mathcal{L}_V$ . Often, it is useful to interpret the difference between the two values,  $dps(h) - dsp(h)$ , as the *degree of ignorance* involved in the judgment of  $h$ . The particular case of  $dsp(h) = 0$  and  $dps(h) = 1$  represents thus a situation of total ignorance with respect to  $h$ , whereas

$dsp(h) = dps(h)$  implies minimal ignorance, a situation in which probabilistic argumentation degenerates into classical probabilistic reasoning. The proper distinction between uncertainty and ignorance is one of the most appealing characteristics of probabilistic argumentation.

For the problem of computing probabilities of DNFs, we will assume that the probability distribution  $\mathbf{P}(W)$  is given by a set  $\Pi = \{p(x) : x \in W\}$  of independent marginal probabilities. The so-called *inclusion-exclusion formula* provides then a mathematically sound but very inefficient solution. More sophisticated methods are known in the domains of *reliability theory* [3, 4, 24] and *knowledge compilation* [12]. For further information on this we refer to the corresponding literature, in particular to Darwiche's d-DNNF compiler [11], which we consider the state-of-the-art in the field.

### 1.3. Goals and Overview

The goal of this paper is to define a general model for distributed trust and authenticity management. The key concepts are the notions of *credentials* and *credential networks*. Section 2 is devoted to the former and Section 3 to the latter. In Subsection 3.1, we will see that many existing models (PGP's web of trust, Haenni's model, Maurer's model, reputation networks, etc.) are included as special cases. How to evaluate credential networks with the aid of probabilistic argumentation is the subject of Section 4. The paper ends with some concluding remarks in Section 5.

## 2. Credentials

The model proposed in this paper is supposed to represent the particular view of a person  $X_0$ , a user of a distributed network. Let  $\mathcal{U}$  denote the set of other network users in which  $X_0$  is interested in. With  $\mathcal{U}_0 = \mathcal{U} \cup \{X_0\}$  we denote the respective set of users that includes  $X_0$ . Note that  $\mathcal{U}_0$  is often a small subset of all users in the network. As we will see in Section 3, the elements of  $\mathcal{U}_0$  will correspond to the nodes of the *credential network* owned by  $X_0$ . We will thus refer to  $X_0$  as the *owner* of the network.

Suppose  $X_0$  wants to use a service offered by another network user  $Y \in \mathcal{U}$ . As noted before, the problem is that user  $Y$  may be unknown to  $X_0$ . In other words,  $X_0$  may not know whether  $Y$  is a trustworthy service provider or not. Moreover,  $X_0$  cannot even be sure if the available information about  $Y$ 's identity is authentic. To formally describe  $X_0$ 's view on  $Y$ , we will use two propositions  $Trust_Y$  and  $Aut_Y$  that are either true or false. Intuitively, the meaning of  $Trust_Y$  is that  $Y$  is trustworthy, whereas  $Aut_Y$  means that the available information about  $Y$ 's identity is authentic (more precise definitions will be given in the following sub-

section). The problem then is to judge whether the propositions  $Trust_Y$  and  $Aut_Y$  are true or false.

If we assume that  $Y \in \mathcal{U}$  is unknown to  $X_0$ , then there is no direct or explicit way of proving  $Trust_Y$  or  $Aut_Y$  to be true or false. The judgment of  $X_0$  must therefore rely upon statements about  $Y$ 's trustworthiness and authenticity issued by third parties, i.e. by other users  $X \in \mathcal{U} \setminus \{Y\}$  of the network. Examples of statements issued by such a user  $X$  are: “I am 90% sure that  $Y$  is trustworthy”, “I know that  $Y$  is not the one she/he claims to be”, “in a scale between 0 and 10, I would rate  $Y$ 's trustworthiness with 7”, and so on. We will make a general distinction between two classes T and A of credentials, depending on whether it is a statement about the trustworthiness or the authenticity of the recipient.

## 2.1. Various Types of Credentials

Statements like the ones mentioned above are called *credentials*. In Section 3, we will suppose the owner  $X_0$  to possess a set  $\mathcal{C}$  of credentials, on which the evaluation of the propositions  $Trust_Y$  and  $Aut_Y$  will be based. A credential is always issued by a user  $X \in \mathcal{U}_0$ , the *issuer*, and concerns another user  $Y \in \mathcal{U}_0$ , the *recipient*. Our convention is to denote A-credentials issued by  $X$  for  $Y$  by  $A_{XY}$  and T-credentials by  $T_{XY}$ . Note that the owner  $X_0$  may as well issue and receive credentials, and we do also not prevent any user  $X \in \mathcal{U}_0$  to issue self-credentials  $A_{XX}$  or  $T_{XX}$ .

In this paper, we will always assume that issuing a credential means to digitally sign its contents. Note that the use of digital signatures is an important security requirement. Otherwise, for example by issuing credentials in the name of another user, faking credentials or tampering in general would be an easy task. The need of digital signatures itself requires every network user  $X \in \mathcal{U}_0$  to possess a digital key pair consisting of a private key  $K_X^{Pr}$  and a public key  $K_X^{Pu}$ . The private key is used to generate a digital signature  $S$ , whereas the verification of  $S$  is done with the aid of the corresponding public key. Of course, private and public keys may also be used to ensure confidentiality of any network communication between two users using public-key en-/decryption.

In such a framework, the proposition  $Aut_Y$ , i.e. to say that user  $Y$  is authentic, gets a more precise meaning, namely that a certain public key  $K_Y^{Pu}$  belongs to a certain user  $Y \in \mathcal{U}_0$  [32]. Accordingly, one can think of an A-credential as a statement about such a binding between a public key and a user, and one should rather speak about the authenticity of a public key instead of the authenticity of the corresponding user. Implicitly, we assume the owner's own public key to be authentic, i.e. the proposition  $Aut_{X_0}$  is always true.

At this point, we also need to state more precisely the

meaning of the proposition  $Trust_Y$ . Intuitively, it states that  $Y$  behaves in a reliable way, but since there are many different things  $Y$  can or may do (offer a service, issue various types of credentials, etc.), it may be possible that  $Y$  is reliable in doing one thing but not the other. To take this into account, Maurer distinguishes different propositions  $Trust_Y^i$  [32]. A more complete attempt to classify various forms of trust has been proposed in [16, 27]. In this paper, we will make the assumption that  $Trust_Y$  is a general proposition about anything  $Y$  may do. Again, we suppose  $Trust_{X_0}$  to be implicitly true for the owner  $X_0$ .

In addition to the two classes A and T, we make a further distinction between three different *signs* +, −, and ± for *positive*, *negative*, and *mixed* credentials, respectively.<sup>3</sup> Intuitively, the idea here is that the issuer of a credential may either want to make a positive or a negative statement about the trustworthiness and authenticity of another network user. A mixed statement can be regarded as a combined rating. The meaning of these signs will be further discussed in the following subsections. In total, we will distinguish six different credential types, as shown by the following table:

		Class	
		A	T
Sign	+	Certificate	Recommendation
	−	Revocation	Discredit
	±	Authenticity Rating	Trust Rating

Another feature of our model allows the issuer  $X$  of a credential  $C_{XY}$ , by assigning a value  $\pi \in [0, 1]$ , to specify the *weight* of the credential.  $\pi = 0$  and  $\pi = 1$  are the two extreme cases of minimal and maximal weight, respectively. Formally, we can therefore define a credential  $C$  as a 5-tuple

$$C = (class, sign, issuer, recipient, weight)$$

with

$$\begin{aligned} class &\in \{T, A\}, \\ sign &\in \{+, -, \pm\}, \\ issuer, recipient &\in \mathcal{U}_0, \\ weight &\in [0, 1]. \end{aligned}$$

To distinguish between the two classes, and to make the formal notation more compact, our convention is to denote A-credentials by

$$A_{issuer,recipient}^{sign,weight} = (A, sign, issuer, recipient, weight)$$

<sup>3</sup>In Jøsang's opinion-based trust model [25], the three credential signs correspond to the three edges of the opinion triangle. For reasons of simplicity, we will not consider credentials that correspond to general opinions in this paper. Note that this is not a conceptual limitation of our approach.

and T-credentials by

$$T_{\text{issuer,recipient}}^{\text{sign,weight}} = (\text{T}, \text{sign}, \text{issuer}, \text{recipient}, \text{weight}).$$

This means that  $A_{XY}^{+0.8} = (\text{A}, +, X, Y, 0.8)$  is an example of a positive A-credential (certificate) of weight 0.8, whereas  $T_{XY}^{\pm 0.7} = (\text{T}, \pm, X, Y, 0.7)$  is a mixed T-credential (trust rating) of weight 0.7. Both of them are issued by  $X$  and concern the same recipient  $Y$ .

Corresponding sets of A- and T-credentials will be denoted by  $\mathcal{A}$  and  $\mathcal{T}$ , respectively. Furthermore, we will use  $\mathcal{A}^+$ ,  $\mathcal{A}^-$ , and  $\mathcal{A}^\pm$  to denote sets of positive, negative, and mixed A-credentials, respectively, and  $\mathcal{T}^+$ ,  $\mathcal{T}^-$ , and  $\mathcal{T}^\pm$  for sets of T-credentials. If  $\mathcal{C}$  is an arbitrary set of credentials, then this notational convention allows us to write

$$\mathcal{C} = \mathcal{A} \cup \mathcal{T} = \mathcal{A}^+ \cup \mathcal{A}^- \cup \mathcal{A}^\pm \cup \mathcal{T}^+ \cup \mathcal{T}^- \cup \mathcal{T}^\pm$$

for the decomposition of  $\mathcal{C}$  into corresponding subsets. In the following subsections, we will discuss the respective meaning of the elements of these subsets more closely.

## 2.2. Certificates and Recommendations

Elements of  $\mathcal{A}^+$ , i.e. positive A-credentials, are called *certificates*. If  $X$  issues a certificate  $A_{XY}^{+\pi}$  of weight  $\pi$  for recipient  $Y$ , it means that  $X$  acts as a guarantor for the authenticity of  $Y$ 's public key. The numerical parameter  $\pi$  expresses  $X$ 's personal confidence in issuing a certificate for  $Y$ . A certificate  $A_{XY}^{+1}$  of maximal weight  $\pi = 1$  is called *absolute*. Note that issuing a certificate  $A_{XY}^{+0}$  of minimal weight  $\pi = 0$  is like issuing no certificate at all.

The logic behind certificates is analogue to Maurer's model [32]. Let  $A_{XY}^{+\pi}$  be a proposition that stands for the event that the certificate  $A_{XY}^{+\pi}$  holds. Because  $X$  may only be partially confident in  $A_{XY}^{+\pi}$ , one can think of it as a random event with a given prior probability  $p(A_{XY}^{+\pi}) = \pi$ . In order to use  $A_{XY}^{+\pi}$  to logically prove  $Aut_Y$ , it is necessary that  $Aut_X$ ,  $Trust_X$ , and  $A_{XY}^{+\pi}$  all happen to be true at the same time. In the same way as in Maurer's model [32], we can thus translate  $A_{XY}^{+\pi}$  into the following logical statement:

$$Aut_X \wedge Trust_X \wedge A_{XY}^{+\pi} \rightarrow Aut_Y. \quad (4)$$

In a similar way, we call positive T-credentials *recommendations*. An element of  $T_{XY}^{+\pi} \in \mathcal{T}^+$  represents thus  $X$ 's opinion that  $Y$  is trustworthy. Again,  $\pi = 1$  means that the recommendation is absolute, whereas  $\pi = 0$  implies that  $X$  has no opinion at all, which is equivalent to not issuing a recommendation.

Let  $T_{XY}^{+\pi}$  be the proposition representing the event that a given recommendation  $T_{XY}^{+\pi}$  holds. The uncertainty involved is again expressed by  $p(T_{XY}^{+\pi}) = \pi$ . The logic behind recommendations is similar as above, i.e. to prove the

trustworthiness of  $Y$ , it is necessary that  $Aut_X$ ,  $Trust_X$ , and  $T_{XY}^{+\pi}$  are all true:

$$Aut_X \wedge Trust_X \wedge T_{XY}^{+\pi} \rightarrow Trust_Y. \quad (5)$$

So far, our model conforms to Maurer's model, except that only one level of trust and recommendations is considered here. Note that this restriction is not a conceptual limitation, but it helps to keep things reasonably simple.

## 2.3. Revocations and Discredits

With respect to trustworthiness and authenticity, the evidence provided by certificates and recommendations is always positive. We will now consider the case of *negative* evidence in the form of *revocations* and *discredits*. Intuitively, a revocation is a signed statement that somebody's public key is *not* authentic. In the literature, it is usually supposed that revocations are either issued by a central authority or by the owner of a key, e.g. in the case of a lost or compromised private key, but here we let all network users be potential issuers of revocations. Another different view is to suppose revocations to work on certificates [36], but here they will always refer directly to the authenticity of a public key.

Formally, a revocation issued by  $X$  for  $Y$  is a negative A-credential  $A_{XY}^{-\pi} \in \mathcal{A}^-$ . Because revocations can be regarded as negative certificates, we use the propositional symbol  $A_{XY}^-$  for the event that the revocation holds. Again, we interpret the weight  $\pi$  as a probability  $p(A_{XY}^-) = \pi$ , and the logic consists of an inference rule in which  $Aut_X$ ,  $Trust_X$ , and  $A_{XY}^-$  imply that  $Aut_Y$  is false:

$$Aut_X \wedge Trust_X \wedge A_{XY}^- \rightarrow \neg Aut_Y. \quad (6)$$

In a similar way, one can think of a statement that somebody is untrustworthy. Such a negative recommendation  $T_{XY}^{-\pi} \in \mathcal{T}^-$  is called *discredit*, and we use the propositional symbol  $T_{XY}^-$  to denote the event that the discredit holds with probability  $p(T_{XY}^-) = \pi$ . The logic behind discredits is analogue to revocations, i.e. if  $Aut_X$ ,  $Trust_X$ , and  $T_{XY}^-$  are all true, then it follows that  $Trust_Y$  is false:

$$Aut_X \wedge Trust_X \wedge T_{XY}^- \rightarrow \neg Trust_Y. \quad (7)$$

In the literature, negative evidence has not yet been studied in depth. Maurer's model does not enable negative evidence at all [32]. Revocations have been proposed by Haenni [22], but discredits seem to be a relatively new concept.

## 2.4. Ratings

Another relatively new concept in the context of public keys are *ratings*. Again, depending on what it is related to, we

need to distinguish between *authenticity ratings* and *trust ratings*, respectively. The idea is that the issuer of a trust rating, for example, may want to rate somebody's trustworthiness on a continuous scale with *fully trustworthy* and *completely untrustworthy* at the two extreme ends. Here, the weight specified by the issuer will therefore play the role of a *degree of trustworthiness* (of the recipient), rather than a confidence value (of the issuer). In practice, trust ratings are probably more useful than authenticity ratings, but we will consider both types.

In the context of this paper, an authenticity rating issued by  $X$  for recipient  $Y$  is a mixed A-credential  $A_{XY}^{\pm\pi} \in \mathcal{A}^{\pm}$  of weight  $\pi$ . The corresponding uncertain event is again denoted by  $A_{XY}^{\pm}$  with  $p(A_{XY}^{\pm}) = \pi$  and  $p(\neg A_{XY}^{\pm}) = 1 - \pi$ . Under the condition that both  $Aut_X$  and  $Trust_X$  are true, the idea is that  $A_{XY}^{\pm}$  implies  $Aut_Y$  and vice versa:

$$Aut_X \wedge Trust_X \rightarrow (A_{XY}^{\pm} \leftrightarrow Aut_Y) \quad (8)$$

A different but logically equivalent form to write this rule is to split it on two separate lines, namely by

$$\begin{aligned} Aut_X \wedge Trust_X \wedge A_{XY}^{\pm} &\rightarrow Aut_Y, \\ Aut_X \wedge Trust_X \wedge \neg A_{XY}^{\pm} &\rightarrow \neg Aut_Y. \end{aligned}$$

This is like combining a certificate and a revocation, linked by the same propositional symbol. This is why ratings are called *mixed* credentials.

In an analogue way, trust ratings are mixed T-credentials  $T_{XY}^{\pm\pi} \in \mathcal{T}^{\pm}$  of weight  $\pi$ . If  $T_{XY}^{\pm}$  with  $p(T_{XY}^{\pm}) = \pi$  denotes the corresponding random event, then

$$Aut_X \wedge Trust_X \rightarrow (T_{XY}^{\pm} \leftrightarrow Trust_Y) \quad (9)$$

describes the logic of trust ratings. In the literature, ratings are often used to build up reputation networks [42], but most authors do not require them to be digitally signed. We have already pointed out that this is a serious security risk.

### 3. Credential Networks

In the previous section, we have proposed six different types of credentials. Suppose now that the users of a distributed network issue such credentials whenever possible. For the distribution of these credentials, they may either put them on respective servers or exchange them individually (e.g. by email attachments). In this way, we suppose  $X_0$  to collect a set  $\mathcal{C} = \mathcal{A} \cup \mathcal{T}$  of credentials concerning a set  $\mathcal{U}_0$  of users. Note that  $\mathcal{C}$  may include credentials that have been issued by  $X_0$ .

In such a context, the *credential network*  $\mathcal{N}$  owned by user  $X_0$  is defined as a 4-tuple

$$\mathcal{N} = (\mathcal{U}_0, X_0, \mathcal{A}, \mathcal{T}).$$

In the following, for a pair of users  $X, Y \in \mathcal{U}_0$ , we will restrict each set  $\mathcal{A}$  and  $\mathcal{T}$  to include at most one credential  $A_{XY} \in \mathcal{A}$  and  $T_{XY} \in \mathcal{T}$ , respectively. Note that this does not exclude cases in which both sets  $\mathcal{A}$  and  $\mathcal{T}$  include a credential between the issuer  $X$  and the recipient  $Y$ . In other words,  $X$  may issue at most one A-credential for  $Y$ , but at the same time,  $X$  may also issue (at most) one T-credential for  $Y$ .

#### 3.1. Examples

We will give now two examples of simple credential networks. The networks are depicted in Fig. 1 and 2 as weighted, directed and (possibly) cyclic multigraphs (i.e. multiple edges between nodes are permitted). Our notational convention is the following:

1. Every circled node denotes a user. The owner is indicated by a double circle.
2. An arrow from  $X$  to  $Y$  denotes a credential issued by user  $X$  for user  $Y$ . Solid arrows are A-credentials (certificates, revocations, authenticity ratings), whereas dotted arrows denote T-credentials (recommendations, discredits, trust ratings).
3. The sign and number attached to an arrow indicate the sign and the weight of the credential.

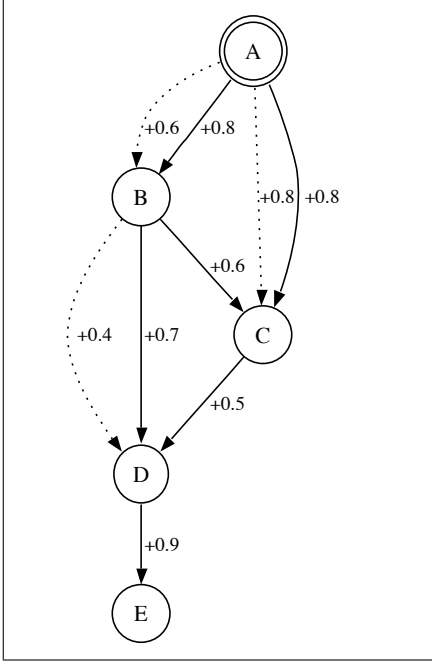
EXAMPLE 3.1. The first credential network is depicted in Fig.1. It consists of five users, six certificates, and three recommendations, i.e. its evidence is purely positive. Formally, we can describe the network as a tuple  $\mathcal{N} = (\mathcal{U}_0, X_0, \mathcal{A}, \mathcal{T})$  with:

$$\begin{aligned} \mathcal{U}_0 &= \{A, B, C, D, E\}, \\ X_0 &= A, \\ \mathcal{A} &= \{A_{AB}^{+0.8}, A_{AC}^{+0.8}, A_{BC}^{+0.6}, A_{BD}^{+0.7}, A_{CD}^{+0.5}, A_{DE}^{+0.9}\}, \\ \mathcal{T} &= \{T_{AB}^{+0.6}, T_{AC}^{+0.8}, T_{BD}^{+0.4}\}. \end{aligned}$$

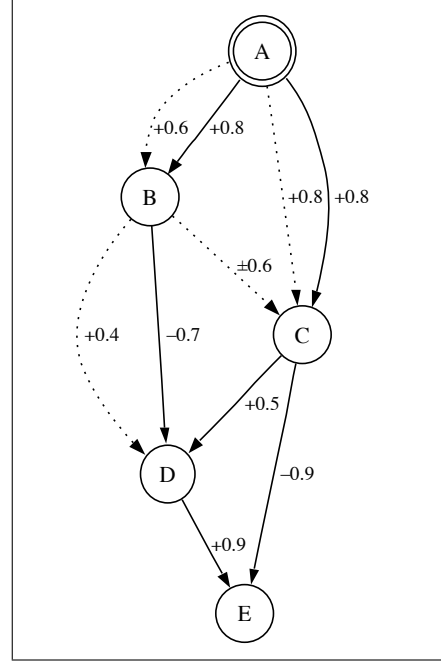
The solid arrow from node  $A$  to node  $B$  represents a certificate of weight 0.8 issued by user  $A$  for recipient  $B$ . Similarly, the dotted arrow from  $A$  to  $B$  is a recommendation of weight 0.6 issued by  $A$  for  $B$ , and so on.

EXAMPLE 3.2. The second example is similar to the first one, but it also includes negative and mixed credentials. It is depicted in Fig. 2 and is formally described by

$$\begin{aligned} \mathcal{U}_0 &= \{A, B, C, D, E\}, \\ X_0 &= A \\ \mathcal{A} &= \{A_{AB}^{+0.8}, A_{AC}^{+0.8}, A_{BD}^{-0.7}, A_{CD}^{+0.5}, A_{CE}^{-0.9}, A_{DE}^{+0.9}\}, \\ \mathcal{T} &= \{T_{AB}^{+0.6}, T_{AC}^{+0.8}, T_{BC}^{\pm 0.6}, T_{BD}^{+0.4}\}. \end{aligned}$$



**Figure 1.** A network with positive credentials only.



**Figure 2.** A network with positive, negative, and mixed credentials.

Instead of a certificate, as it was the case in the first example, user  $B$  is now the issuer of a trust rating of weight  $0.6$  for recipient  $C$ . Similarly,  $B$ 's certificate for  $D$  is now a revocation of the same weight  $0.7$ . Another new revocation of weight  $0.9$  is the one between  $C$  and  $E$ .

### 3.2. Existing Models

From the perspective of credential networks, let us now discuss the relation to existing models. Maurer's model allows certificates and recommendations with corresponding confidence values, but no negative or mixed statements [32]. If we neglect Maurer's distinction between different trust levels (see remark in Subsection 2.2), we get credential networks of the following form:

$$\mathcal{N}_{\text{Maurer}} = (\mathcal{U}_0, X_0, \mathcal{A}^+, \mathcal{T}^+).$$

In Haenni's model [22], all certificates are absolute. Furthermore, the owner  $X_0$  is required to specify explicit trust values for all users in the network. This is like requiring the owner to issue trust ratings for all users. Let  $\mathcal{T}_{X_0\mathcal{U}}^\pm$  denote such a set of trust ratings issued by  $X_0$ , i.e. one for each user  $X \in \mathcal{U}$ . If we consider (absolute) revocations to be part of Haenni's model, but not recommendations, then this corresponds to

$$\mathcal{N}_{\text{Haenni}} = (\mathcal{U}_0, X_0, \mathcal{A}^+ \cup \mathcal{A}^{-1}, \mathcal{T}_{X_0\mathcal{U}}^\pm)$$

The situation in a PGP web of trust is similar to Haenni's model. All certificates are absolute, and the owner of the

web of trust is required to specify trust values for all users. However, because the possible trust values are restricted to *completely trustworthy*, *marginally trustworthy*, and *untrustworthy*, it's like restricting the possible weights of the trust ratings to  $\{0, \frac{1}{2}, 1\}$ . A set of such trust ratings for all users in  $\mathcal{U}$  is denoted by  $\mathcal{T}_{X_0\mathcal{U}}^{\pm\{0, \frac{1}{2}, 1\}}$ . If we do not consider revocations to be part of the PGP model (PGP allows only self-revocations), then we get the following particular kind of credential network:

$$\mathcal{N}_{\text{PGP}} = (\mathcal{U}_0, X_0, \mathcal{A}^{+1}, \mathcal{T}_{X_0\mathcal{U}}^{\pm\{0, \frac{1}{2}, 1\}})$$

In a centralized model, it is assumed that (at least) one network user  $X_{CA}$  is a fully trustworthy *certificate authority*. All certificates are absolute, and with one exception they are all issued by  $X_{CA}$ . The one that is not issued by  $X_{CA}$  is a certificate issued by  $X_0$  for  $X_{CA}$ . Let  $\mathcal{U}_{CA} = \mathcal{U}_0 \cup \{X_{CA}\}$  be the extended set of users that includes a certificate authority  $X_{CA}$ . If  $\mathcal{A}_{X_{CA}}^{+1}$  denotes a set of absolute certificates all issued by  $X_{CA}$ , then a centralized model is a credential network with the following characteristics:

$$\mathcal{N}_{\text{Centralized}} = (\mathcal{U}_{CA}, X_0, \mathcal{A}_{X_{CA}}^{+1} \cup \{\mathcal{A}_{X_0 X_{CA}}^{+1}\}, \{\mathcal{T}_{X_0 X_{CA}}^{+1}\}).$$

In reputation networks, all the statements are ratings, but they are usually not required to be digitally signed. Furthermore, the focus is on trust only. In the terminology of credential networks, this is like assuming all public keys to

be authentic, which corresponds to a situation in which  $X_0$  issues absolute certificates for all users  $X \in \mathcal{U}$ . If  $\mathcal{A}_{X_0\mathcal{U}}^{+1}$  denotes such a set of absolute certificates issued by  $X_0$ , then

$$\mathcal{N}_{\text{Reputation-Networks}} = (\mathcal{U}_0, X_0, \mathcal{A}_{X_0\mathcal{U}}^{+1}, \mathcal{T}^{\pm})$$

forms a reputation network.

## 4. Evaluation

For a given credential network, the question that arises now is how to evaluate it. On the basis of the evidence encoded in the network, and with respect to a particular user  $X \in \mathcal{U}$ , the primary goal is to quantitatively judge

- (1) the authenticity (of a given public key) of  $X$  and
- (2) the trustworthiness of  $X$ .

The judgment should return corresponding values on a continuous scale between 0 and 1. The owner of the credential network may then use this information to decide whether a public key or a service should be accepted or not.

The approach we propose is to translate the credential network into a probabilistic argumentation system (see Subsection 1.2), and then to compute degrees of support and possibility for the hypotheses  $Aut_X$  and  $Trust_X$ . In the following subsections, we will describe such a translation and the subsequent evaluation. The core is a special purpose algorithm that generates sets of minimal arguments for both  $Aut_X$  and  $Trust_X$  for all users  $X \in \mathcal{U}_0$ .

### 4.1. From Credential Networks to Probabilistic Argumentation Systems

Consider a credential network  $\mathcal{N} = (\mathcal{U}_0, X_0, \mathcal{A}, \mathcal{T})$  owned by  $X_0$ , and suppose that  $\mathcal{U}_0 = \{X_0, X_1, \dots, X_n\}$  is the set of all users and  $\mathcal{C} = \mathcal{A} \cup \mathcal{T} = \{C_1, \dots, C_m\}$  the set of all credentials involved. To build a corresponding probabilistic argumentation system  $\mathcal{S} = (V, W, \mathbf{P}, \Sigma)$ , we need to define the two sets of propositions  $V$  and  $W$ , the probability distribution  $\mathbf{P}$ , and the set  $\Sigma$  of propositional sentences.

From the logical descriptions of the six different credential types, see (4) to (9) in Subsection 2.2 to 2.4, it follows that  $\Sigma$  will essentially consist of  $m$  logical inference rules, i.e. one for each credential. Let  $\gamma(C)$  be the corresponding rule for a particular credential  $C \in \mathcal{C}$ . Because  $Aut_{X_0}$  and  $Trust_{X_0}$  are implicitly true, we have to consider a total number of  $m + 2$  propositional sentences:

$$\Sigma = \{Aut_{X_0}, Trust_{X_0}\} \cup \{\gamma(C) : C \in \mathcal{C}\}. \quad (10)$$

To illustrate this, consider the credential network depicted

in Fig. 1, from which we obtain:

$$\Sigma = \left\{ \begin{array}{l} Aut_A \\ Trust_A \\ Aut_A \wedge Trust_A \wedge A_{AB}^+ \rightarrow Aut_B \\ Aut_A \wedge Trust_A \wedge A_{AC}^+ \rightarrow Aut_C \\ Aut_B \wedge Trust_B \wedge A_{BC}^+ \rightarrow Aut_C \\ Aut_B \wedge Trust_B \wedge A_{BD}^+ \rightarrow Aut_D \\ Aut_C \wedge Trust_C \wedge A_{CD}^+ \rightarrow Aut_D \\ Aut_D \wedge Trust_D \wedge A_{DE}^+ \rightarrow Aut_E \\ Aut_A \wedge Trust_A \wedge T_{AB}^+ \rightarrow Trust_B \\ Aut_A \wedge Trust_A \wedge T_{AC}^+ \rightarrow Trust_C \\ Aut_B \wedge Trust_B \wedge T_{BD}^+ \rightarrow Trust_D \end{array} \right\}.$$

The propositions involved are  $Aut_X$  and  $Trust_X$  for all users  $X \in \mathcal{U}_0$ ,  $A_{XY}^+$  for all certificates  $A_{XY}^{+\pi} \in \mathcal{A}$ ,  $T_{XY}^+$  for all recommendations  $T_{XY}^{+\pi} \in \mathcal{T}$ ,  $A_{XY}^-$  for all revocations  $A_{XY}^{-\pi} \in \mathcal{A}$ , and so on. A probability distribution  $\mathbf{P}$  is implicitly given over the propositions  $A_{XY}^+$ ,  $T_{XY}^+$ ,  $A_{XY}^-$ , etc., namely by the corresponding marginal probabilities  $p(A_{XY}^+)$ ,  $p(T_{XY}^+)$ ,  $p(A_{XY}^-)$ , etc. As stated before, we assume them to be probabilistically independent. The sets  $W$  and  $V$  are therefore defined by

$$\begin{aligned} W = & \{A_{XY}^+ : A_{XY}^{+\pi} \in \mathcal{A}\} \cup \{T_{XY}^+ : T_{XY}^{+\pi} \in \mathcal{T}\} \cup \\ & \{A_{XY}^- : A_{XY}^{-\pi} \in \mathcal{A}\} \cup \{T_{XY}^- : T_{XY}^{-\pi} \in \mathcal{T}\} \cup \\ & \{A_{XY}^{\pm} : A_{XY}^{\pm\pi} \in \mathcal{A}\} \cup \{T_{XY}^{\pm} : T_{XY}^{\pm\pi} \in \mathcal{T}\}, \end{aligned}$$

$$V = W \cup \{Aut_X : X \in \mathcal{U}_0\} \cup \{Trust_X : X \in \mathcal{U}_0\}.$$

Based on such a probabilistic argumentation system  $\mathcal{S} = (V, W, \mathbf{P}, \Sigma)$ , we will now evaluate the corresponding credential network for all hypotheses  $Aut_X$  and  $Trust_X$  of interest.

### 4.2. Qualitative and Quantitative Evaluation

A *qualitative* evaluation of a probabilistic argumentation system means finding minimal sets of arguments, counter-arguments, and conflicts (see Subsection 1.2). In the context of the problem addressed in this paper, it is possible to interpret a minimal argument  $\alpha \in \text{args}(Aut_X)$ , for example, as a directed certificate path or chain in the network from  $X_0$  to  $X$ . Additionally, each node  $Y$  along such a path needs to be recursively supported by a corresponding recommendation path from  $X_0$  to  $Y$ , and so on (see [22] for further details).

From a *quantitative* point of view, we are not interested in arguments themselves, but in corresponding degrees of support and possibility. These values result from the weights attached to the credentials involved in the network. To decide whether a hypothesis  $Trust_X$  or  $Aut_X$  is

accepted or not, the owner may define a threshold  $\lambda \in [0, 1]$  and accept  $Aut_X$  whenever  $dsp(Aut_X) \geq \lambda$  and  $Trust_X$  whenever  $dsp(Trust_X) \geq \lambda$ . Note that not accepting a hypothesis is not necessarily a reason to reject it. For this, it may be necessary to define another threshold  $\eta \in [0, 1]$ , i.e.  $Aut_X$  is rejected for  $dps(Aut_X) \leq \eta$  and  $Trust_X$  for  $dps(Trust_X) \leq \eta$ . If a hypothesis is neither accepted nor rejected, it means that the available information is insufficient to make a decision, i.e. more credentials are needed.

EXAMPLE 4.1. Consider again the example shown in Fig. 1 and suppose that we are interested in the authenticity and trustworthiness of user  $D$ . It turns out that there are three minimal arguments supporting the hypothesis  $Aut_D$ , namely

$$args(Aut_D) = \left\{ \begin{array}{l} A_{AB}^+ A_{BD}^+ T_{AB}^+ \\ A_{AC}^+ A_{CD}^+ T_{AC}^+ \\ A_{AB}^+ A_{BC}^+ A_{CD}^+ T_{AB}^+ T_{AC}^+ \end{array} \right\},$$

but there is only one minimal argument for  $Trust_D$ :

$$args(Trust_D) = \{A_{AB}^+ T_{AB}^+ T_{BD}^+\}.$$

Because the example consists of positive evidence only, there are no counter-arguments for  $Aut_D$  or  $Trust_D$ , i.e. the corresponding sets  $args(\neg Aut_D) = args(\neg Trust_D) = \emptyset$  are both empty. This is also true for all other users, and it implies that the set of conflicts  $args(\perp, \Sigma) = \emptyset$  is also empty.

From a *quantitative* point of view, we are not interested in arguments themselves, but in corresponding degrees of support and possibility. These values result from the weights attached to the credentials involved in the network. In our conflict-free example, it follows from (3) that  $dps(Aut_X) = dsp(Trust_X) = 1$  for all users  $X \in \mathcal{U}_0$ . So the only relevant quantity here is degree of support. Note that  $args(\perp, \Sigma) = \emptyset$  implies  $p(args(\perp)) = 0$ , which allows us to simplify (2) applied to  $Aut_X$  and  $Trust_X$  to

$$\begin{aligned} dsp(Aut_X) &= p(args(Aut_X)), \\ dsp(Trust_X) &= p(args(Trust_X)), \end{aligned}$$

for all  $X \in \mathcal{U}_0$ . The following table shows the corresponding degrees of support and possibility for all users  $A$  to  $E$  in the network of Fig. 1.

	User $X$				
	$A$	$B$	$C$	$D$	$E$
$dsp(Aut_X)$	1	0.8	0.858	0.555	0.14
$dps(Aut_X)$	1	1	1	1	1
$dsp(Trust_X)$	1	0.6	0.8	0.192	0
$dps(Trust_X)$	1	1	1	1	1

The owner  $A$  receives of course maximal support for both authenticity and trust. On the other hand, the trustworthiness of  $E$  is completely unsupported, because nobody has issued a recommendation for  $E$ . All other values lie somewhere between 0 and 1. Suppose  $A$ 's threshold for accepting a hypothesis is  $\lambda = 0.8$ . In this case, only  $Aut_B$ ,  $Aut_C$ , and  $Trust_C$  would be accepted. Note that  $A$  has no reason to reject anything, i.e. all other hypotheses should be left open until more evidence allows a more precise judgment.

EXAMPLE 4.2. Now have a look at the second example shown in Fig. 2. The situation is a bit more complicated, because the sets of counter-arguments and conflicts are no longer empty. This is as a consequence of the negative and mixed credentials involved. For example,  $A_{AB}^+ A_{BD}^- T_{AB}^+$  is a counter-argument for  $Aut_D$ . Together with the argument  $A_{AC}^+ A_{CD}^+ T_{AC}^+$  for  $Aut_D$ , we obtain a conflict  $A_{AB}^+ A_{AC}^+ A_{BD}^- A_{CD}^+ T_{AB}^+ T_{AC}^+$ .

In general, if the sets  $args(Aut_X)$  and  $args(\neg Aut_X)$  are non-empty for some  $X \in \mathcal{U}$ , then conflicts are constructed by conjoining corresponding pairs of minimal arguments and counter-arguments. In our example, the complete set of minimal conflicts is

$$args(\perp) = \left\{ \begin{array}{l} A_{AB}^+ T_{AB}^+ T_{AC}^+ \neg T_{BC}^\pm \\ A_{AB}^+ A_{AC}^+ A_{BD}^- A_{CD}^+ T_{AB}^+ T_{AC}^+ \\ A_{AB}^+ A_{AC}^+ A_{BD}^- A_{CD}^+ T_{AB}^+ T_{BC}^\pm \\ A_{AB}^+ A_{AC}^+ A_{CD}^+ A_{CE}^- A_{DE}^+ T_{AB}^+ T_{AC}^+ T_{BD}^+ \\ A_{AB}^+ A_{AC}^+ A_{CD}^+ A_{CE}^- A_{DE}^+ T_{AB}^+ T_{BC}^\pm T_{BD}^+ \end{array} \right\}.$$

The presence of counter-arguments and conflicts has of course some impact on the quantitative evaluation of trust and authenticity. In the presence of negative evidence, degrees of support and possibility will normally decrease. The new values for this example are shown in the following table.

	User $X$				
	$A$	$B$	$C$	$D$	$E$
$dsp(Aut_X)$	1	0.735	0.776	0.251	0.002
$dps(Aut_X)$	1	0.919	0.97	0.789	0.439
$dsp(Trust_X)$	1	0.470	0.811	0.115	0
$dps(Trust_X)$	1	0.783	0.949	0.799	1

Suppose the owner  $A$  still uses the same threshold  $\alpha = 0.8$  for accepting a hypothesis and  $\eta = 0.5$  for rejecting a hypothesis. In this case  $Trust_C$  would be accepted and  $Aut_E$  rejected. All other hypotheses are left open.

### 4.3. Computing Arguments

Let us now turn our attention to the computational problem of finding the minimal sets of arguments, counter-arguments, and conflicts. General algorithms are well documented in the literature [18, 19], but here the goal is to construct a special purpose algorithm for the particular type of probabilistic argumentation system resulting from a credential network. Due to the space restrictions of this paper, we will focus the discussion to the conflict-free case of positive evidence obtained from certificates and recommendations. The only problem addressed here is thus the computation of minimal arguments for all hypotheses  $Aut_X$  and  $Trust_X$ .

The core of the algorithm are the following two theorems. They are simple consequences of the inference rules (4) and (5) in Subsection 2.2. In both cases we start from a credential network  $\mathcal{N} = (\mathcal{U}_0, X_0, \mathcal{A}, \mathcal{T})$  and assume that  $\mathcal{N}$  has been translated into a corresponding probabilistic argumentation system  $\mathcal{S} = (V, W, \mathbf{P}, \Sigma)$  as described in Subsection 4.1. The first theorem tells us how to generate (not necessarily minimal) arguments for  $Aut_X$ , whereas the logic for generating arguments for  $Trust_X$  follows from the second theorem. Analogue theorems exist for negative and mixed credentials.

**Theorem 4.1.** *Let  $A_{XY}^+ \in \mathcal{A}$  be a certificate issued by user  $X$  for recipient  $Y$ . If  $\alpha \in \text{Args}(Aut_X)$  and  $\beta \in \text{Args}(Trust_X)$ , then*

$$\alpha \cup \beta \cup \{A_{XY}^+\} \in \text{Args}(Aut_Y).$$

**Theorem 4.2.** *Let  $T_{XY}^+ \in \mathcal{T}$  be a recommendation issued by user  $X$  for recipient  $Y$ . If  $\alpha \in \text{Args}(Aut_X)$  and  $\beta \in \text{Args}(Trust_X)$ , then*

$$\alpha \cup \beta \cup \{T_{XY}^+\} \in \text{Args}(Trust_Y).$$

With the two inference rules (4) and (5) in mind, the proofs of these theorems are straightforward. To show that *all* arguments are obtained in this way is a bit more difficult. In the purely positive case, however, because (4) and (5) and therefore all sentences in  $\Sigma$  are *Horn* clauses, it is possible to prove completeness in the same way as it is usually done in the literature on abduction, theorem proving, or consequence finding in Horn theories [13]. The space limitation of this paper does not allow us to give further information on this.

We will now introduce two recursive procedures  $\text{add\_aut\_arg}(\alpha, X)$  and  $\text{add\_trust\_arg}(\alpha, X)$ , which compute the minimal sets of arguments  $\text{args}(Aut_X)$  and  $\text{args}(Trust_X)$  for all users  $X \in \mathcal{U}_0$ . Let  $\mathcal{U}_X^A \subseteq \mathcal{U}_0$  denote the set of all users in the network for which  $X$  has issued a certificate, and similarly  $\mathcal{U}_X^T \subseteq \mathcal{U}_0$  for the recommendations. In the graph of Fig. 1, for example, we have  $\mathcal{U}_A^A = \{B, C\}$ ,  $\mathcal{U}_A^T = \{B, C\}$ ,  $\mathcal{U}_B^A = \{C, D\}$ ,

$\mathcal{U}_B^T = \{D\}$ , and so on. Furthermore, let  $\text{args}^*(Aut_X)$  and  $\text{args}^*(Trust_X)$  for all users  $X \in \mathcal{U}_0$  be (initially empty) sets of arguments for  $Aut_X$  and  $Trust_X$ , respectively.

The idea of the algorithm is to incrementally fill up these sets according to the above theorems. The sets are always kept minimal, i.e. before a new argument  $\alpha$  is actually added, the algorithm checks whether  $\alpha$  is minimal with respect to the current set or not. If  $\alpha$  is minimal, other non-minimal arguments are deleted and a cascade of further procedure calls with other new arguments is initiated. Otherwise, the recursion stops. Note that both procedures  $\text{add\_aut\_arg}(\alpha, X)$  and  $\text{add\_trust\_arg}(\alpha, X)$  are perfectly symmetric.

```

1 procedure add_aut_arg( $\alpha, X$ )
2 begin
3   if  $\nexists \alpha' \in \text{args}^*(Aut_X)$  such that  $\alpha' \subset \alpha$  then
4      $\text{args}^*(Aut_X) \leftarrow \mu(\text{args}^*(Aut_X) \cup \{\alpha\})$ 
5     for  $\beta \in \text{args}^*(Trust_X)$  do
6       for  $Y \in \mathcal{U}_X^A$  do
7          $\text{add\_aut\_arg}(\alpha \cup \beta \cup \{A_{XY}^+\}, Y)$ 
8       end
9       for  $Y \in \mathcal{U}_X^T$  do
10         $\text{add\_trust\_arg}(\alpha \cup \beta \cup \{T_{XY}^+\}, Y)$ 
11      end
12    end
13  end
14 end

```

```

1 procedure add_trust_arg( $\alpha, X$ )
2 begin
3   if  $\nexists \alpha' \in \text{args}^*(Trust_X)$  such that  $\alpha' \subset \alpha$  then
4      $\text{args}^*(Trust_X) \leftarrow \mu(\text{args}^*(Trust_X) \cup \{\alpha\})$ 
5     for  $\beta \in \text{args}^*(Aut_X)$  do
6       for  $Y \in \mathcal{U}_X^A$  do
7          $\text{add\_aut\_arg}(\alpha \cup \beta \cup \{A_{XY}^+\}, Y)$ 
8       end
9       for  $Y \in \mathcal{U}_X^T$  do
10         $\text{add\_trust\_arg}(\alpha \cup \beta \cup \{T_{XY}^+\}, Y)$ 
11      end
12    end
13  end
14 end

```

The recursive computation of the arguments starts by adding the empty argument  $\emptyset$  (the one that is always true) to both sets for the owner  $X_0$ . This reflects the assumption that  $X_0$  is implicitly authentic and trustworthy. The two initial calls are therefore  $\text{add\_aut\_arg}(\emptyset, X_0)$  and  $\text{add\_trust\_arg}(\emptyset, X_0)$ , and the main procedure looks as follows:

```

1 procedure compute_args ( )
2 begin
3   for  $X \in \mathcal{U}_0$  do
4     |    $args^*(Aut_X) \leftarrow \emptyset$ 
5     |    $args^*(Trust_X) \leftarrow \emptyset$ 
6   end
7   add_aut_arg( $\emptyset, X_0$ )
8   add_trust_arg( $\emptyset, X_0$ )
9 end

```

Once the computation terminates, we obtain the minimal sets of arguments  $args(Aut_X)$  and  $args(Trust_X)$  for all users  $X \in \mathcal{U}_0$  in the credential network, as required.

Note again that this algorithm only deals with certificates and recommendations. For the general case of positive, negative, and mixed credentials, two other analogous procedures  $add\_aut\_counter\_arg(\alpha, X)$  and  $add\_trust\_counter\_arg(\alpha, X)$  are required, and the procedures given here need to be extended accordingly. All these procedures including the necessary extensions have been implemented.<sup>4</sup>

The above algorithm works like a recursive *depth-first* search in a graph. This is the reason why non-minimal arguments are possibly produced during the computation. They are temporarily stored in the respective sets  $args^*(Aut_X)$  and  $args^*(Trust_X)$ , from which they are deleted when a shorter argument is found. A corresponding *breadth-first* search would avoid such non-minimal arguments, but empirical tests have shown that there is no significant difference in the performance between these two approaches. In large networks the depth-first version performs even better, because the length of the queue in the breadth-first implementation becomes problematical.

## 5. Conclusions

This paper suggests a general model for distributed trust and authenticity management. The core of the model is a distinction between six different types of credentials. Every credential is a weighted and digitally signed statement about somebody else's trustworthiness or authenticity. A collection of credentials defines a credential network.

For the evaluation of a credential network, the paper proposes a translation into a probabilistic argumentation systems. For a given credential network, evaluating trust and authenticity is then a problem of computing so-called degrees of support and possibility. For the case of purely

<sup>4</sup>The source code of a program written in LISP is available at <http://www.iam.unibe.ch/~run/trust.html>.

positive evidence, the paper describes an algorithm to compute corresponding sets of arguments, from which degrees of support are derived in a second step.

## Acknowledgements

Special thanks to Michael Wachter for careful proof-reading and comments and to Reto Kohlas for helpful discussions.

## References

- [1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *HICSS-33, 33rd Hawaii International Conference on System Sciences*, pages 1769–1777, Maui, Hawaii, 2000.
- [2] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In H. Paques, L. Liu, and D. Grossman, editors, *CIKM01, 10th International Conference on Information and Knowledge Management*, pages 310–317. ACM Press, 2001.
- [3] J. A. Abraham. An improved algorithm for network reliability. *IEEE Transactions on Reliability*, 28:58–61, 1979.
- [4] B. Anrig. A generalization of the algorithm of Abraham. In M. Nikulin and N. Limnios, editors, *MMR'2000: Second International Conference on Mathematical Methods in Reliability*, pages 95–98, Bordeaux, France, 2000.
- [5] T. Beth, M. Borcherdig, and B. Klein. Valuation of trust in open networks. In *ESORICS'94, 3rd European Symposium on Research in Computer Security*, LNCS 875, pages 3–18. Springer, 1994.
- [6] A. D. Birrell, B. W. Lampson, R. M. Needham, and M. D. Schoreder. A global authentication service without global trust. In *IEEE Symposium on Security and Privacy*, pages 223–230, 1986.
- [7] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *SP'96: IEEE Symposium on Security and Privacy*, pages 164–173, 1996.
- [8] M. Branchaud and S. Flinn. <sup>x</sup>Trust: A scalable trust management infrastructure. In *PST'04: 2nd Annual Conference on Privacy, Security and Trust*, pages 207–218, Fredericton, New Brunswick, Canada, 2004.
- [9] M. Branstad, W. C. Barker, and P. Cochrane. The role of trust in protected mail. In *IEEE Symposium on Security and Privacy*, pages 210–215, 1990.
- [10] D. W. Chadwick, A. J. Young, and N. Kapidzic Cicovic. Merging and extending the PGP and PEM trust models - the ICE-TEL trust model. *IEEE Networks Special Publication on Internet Security*, 11(3):16–24, 1997.
- [11] A. Darwiche. A compiler for deterministic, decomposable negation normal form. In *AAAI'02, 18th National Conference on Artificial Intelligence*, pages 627–634. AAAI Press, 2002.
- [12] A. Darwiche and P. Marquis. A perspective on knowledge compilation. In B. Nebel, editor, *IJCAI'01, 17th International Joint Conference on Artificial Intelligence*, pages 175–182, Seattle, USA, 2001.

- [13] T. Eiter and K. Makino. On computing all abductive explanations. In *AAAI'02: 18th National Conference on Artificial Intelligence*, pages 62–67, Edmonton, Canada, 2002.
- [14] R. Falcone and C. Castelfranchi. Social trust: A cognitive approach. In C. Castelfranchi and Y. H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer Academic Publishers, 2001.
- [15] J. Golbeck and J. Hendler. Reputation network analysis for email filtering. In *CEAS 2004, 1st Conference on Email and Anti-Spam*, Mountain View, CA, 2004.
- [16] T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3(4), 2000.
- [17] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer systems. In *NOSSDAV 2003, 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 144–152, Monterey, CA, 2003.
- [18] R. Haenni. Cost-bounded argumentation. *International Journal of Approximate Reasoning*, 26(2):101–127, 2001.
- [19] R. Haenni. Anytime argumentative and abductive reasoning. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 8(2):142–149, 2003.
- [20] R. Haenni. Web of trust: Applying probabilistic argumentation to public-key cryptography. In *ECSQARU'03, 7th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 243–254, Aalborg, Denmark, 2003.
- [21] R. Haenni. Unifying logical and probabilistic reasoning. In L. Godo, editor, *ECSQARU'05, 8th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, LNAI 3571, pages 788–799, Barcelona, Spain, 2005.
- [22] R. Haenni. Using probabilistic argumentation for key validation in public-key cryptography. *International Journal of Approximate Reasoning*, 38(3):355–376, 2005.
- [23] R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In J. Kohlas and S. Moral, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*, pages 221–288. Kluwer Academic Publishers, 2000.
- [24] K. D. Heidtmann. Statistical comparison of two sum-of-disjoint-product algorithms for reliability and safety evaluation. In *SAFECOMP 2002, 21st International Conference on Computer Safety, Reliability and Security*, pages 70–81. Springer, 2002.
- [25] A. Jøsang. An algebra for assessing trust in certification chains. In *NDSS'99: 6th Annual Symposium on Network and Distributed System Security*, San Diego, USA, 1999.
- [26] A. Jøsang. Trust-based decision making for electronic transactions. In L. Yngström and T. Svensson, editors, *NORDSEC'99: Fourth Nordic Workshop on Secure IT Systems*, Stockholm, Sweden, 1999.
- [27] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 2005 (to appear).
- [28] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *WWW2003, 12th International World Wide Web Conference*, pages 640–651, Budapest, Hungary, 2003.
- [29] M. Kinader and K. Rothermel. Architecture and algorithms for a distributed reputation system. In *iTrust'03: 1st International Conference on Trust Management*, pages 1–16, Heraklion, Greece, 2003.
- [30] R. Kohlas and U. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In H. Imai and Y. Zheng, editors, *PKC'2000, Third International Workshop on Practice and Theory in Public Key Cryptography*, LNCS 1751, pages 93–112, Melbourne, Australia, 2000.
- [31] H. Lei and G. C. Shoja. A distributed trust model for e-commerce applications. In *EEE'05: International Conference on e-Technology, e-Commerce and e-Service*, pages 290–293, Hong Kong, China, 2005.
- [32] U. Maurer. Modelling a public-key infrastructure. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *ESORICS: European Symposium on Research in Computer Security*, LNCS 1146, pages 324–350. Springer, 1996.
- [33] M. A. Patton and A. Jøsang. Technologies for trust in electronic commerce. *Electronic Commerce Research*, 4(1–2):9–21, 2004.
- [34] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, USA, 1988.
- [35] M. K. Reiter and S. G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2):138–158, 1999.
- [36] B. Sadighi Firozabadi and M. Sergot. Revocation in the privilege calculus. In *FAST'03: 1st International Workshop on Formal Aspects in Security and Trust*, pages 39–51, 2003.
- [37] P. Smets. Probability of provability and belief functions. In M. Clarke, R. K. R., and S. Moral, editors, *ECSQARU'93, 2nd European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, LNCS 747, pages 332–340. Springer, 1993.
- [38] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer eCommerce communities. In *CEC'03, IEEE Conference on Electronic Commerce*, pages 275–284, Newport Beach, CA, 2003.
- [39] L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.
- [40] R. Yahalem, B. Klein, and T. Beth. Trust relationships in secure system – a distributed authentication perspective. In *IEEE Symposium on Research in Security and Privacy*, pages 150–164, 1993.
- [41] B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.
- [42] Q. Zhang, T. Yu, and K. Irwin. A classification scheme for trust functions in reputation-based trust management. In *ISWC'04, 3rd International Semantic Web Conference, Workshop on "Trust, Security, and Reputation on the Semantic Web"*, Hiroshima, Japan, 2004.
- [43] P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1994.