Modelling a Public-Key Infrastructure[1]

Ueli Maurer[2]

Department of Computer Science
Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland

Abstract. A global public-key infrastructure (PKI), components of which are emerging in the near future, is a prerequisite for security in distributed systems and for electronic commerce. The purpose of this paper is to propose an approach to modelling and reasoning about a PKI from a user Alice's point of view. Her view, from which she draws conclusions about the authenticity of other entities' public keys and possibly about the trustworthiness of other entities, consists of statements about which public keys she believes to be authentic and which entities she believes to be trustworthy, as well as a collection of certificates and recommendations obtained or retrieved from the PKI. The model takes into account recommendations for the trustworthiness of entities. Furthermore, it includes confidence values for statements and can exploit arbitrary certification structures containing multiple intersecting certification paths to achieve a higher confidence value than for any single certification path. Confidence values are measured on a continuous scale between 0 and 1 and, in contrast to previous work in this area, are interpreted as probabilities in a well-defined random experiment.

Key words. Distributed system security, key management, public-key certification, cryptography, trust, recommendations, probabilistic logic.

1 Introduction

A global public-key infrastructure (PKI) is a prerequisite for security in large networks and distributed systems, and for electronic commerce. While the basic mechanism of public-key certification is well-understood, the problem of building a large distributed PKI is not. The purpose of this paper is to complement previous work on practical ad-hoc approaches to building a PKI by suggesting a precise model of a user's view of a PKI.

In our context, a PKI consists of the entire, generally heterogeneous, set of components that can be involved in issuing, storing, and/or distributing certificates. A PKI can be seen as a distributed database of public-key certificates and

[1] To appear in *Proc. 1996 European Symposium on Research in Computer Security (ESORICS' 96)*, E. Bertino (Ed.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, Rome, Italy, Sept. 1996.

[2] E-mail: maurer@inf.ethz.ch
WWW: http://www.inf.ethz.ch/department/TI/um/group.html

further information (e.g. revocation lists, recommendations, etc.). It provides mechanisms for entities to retrieve and possibly also to add information to the PKI. Typically, an entity Alice can retrieve another entity Bob's public key together with evidence of its authenticity. On the other hand, a user can possibly also contribute to building the infrastructure by certifying other entities' public keys or by issuing recommendations. Such certificates and recommendations can be used by entities who trust Alice but are useless for other entities.

In this paper we are not concerned with distributed database and software aspects of a PKI, i.e., we do not consider the problem of how a user Alice can obtain the necessary certificates to authenticate another user Bob's public key. We rather analyse what kind of procedure Alice could use for deriving conclusions, once she has obtained the necessary information. Of course, these two phases need not be independent; Alice (or her system) may initiate a second phase of collecting evidence when realizing that it needs more information for deriving a certain conclusion. In an implementation, collecting evidence can be based on a number of different mechanisms: accessing an official certificate service, retrieval from the certificate databases distributed over the Internet, or an automated negotiation process between Alice's and Bob's systems by which Bob's system provides the necessary certificates or links to certificates needed by Alice's system.

Whether given information constitutes sufficient evidence for Alice for the authenticity of Bob's public key depends on various parameters to be set by Alice, including her assumptions about the trustworthiness of certificate-issuing entities, the authenticity of certain public keys stored in her own data base, and the security requirements of the particular application in which the public key is going to be used.

In a simple model of public-key certification, a user Alice uses a path (or chain) of certificates where each public key is certified by the previous entity in the path, and where she has specified the first public key as authentic and all intermediate entities as trustworthy. Such a simple model can be insufficient for various reasons. First, in a realistic scenario, it should be possible to assign confidence parameters[1] (for instance between 0 and 1) to statements about authenticity and trust. Second, it should be possible to take into account multiple certification paths which, in general, are not independent but can rather be intersecting paths in a possibly complex directed acyclic graph of certificates. Third, trust is often based on recommendations. For instance, Alice may trust an entity T she does not know because it has been recommended as trustworthy by one or several other entities that Alice trusts.

In our model, conclusions about whether a given public key is sufficiently authenticated to be used in a particular application, are derived from Alice's view. Her view consists of statements about which public keys she believes initially to be authentic and which entities she believes initially to be trustworthy, and a collection of certificates and recommendations obtained or retrieved

[1] We use the term confidence parameter and confidence value when it is assigned by the entities or derived within the model, respectively.

from the PKI. The model takes into account confidence values for statements and can exploit arbitrary certification structures containing multiple intersecting certification paths in order to achieve a higher confidence value than for any single certification path. Confidence values are measured on a continuous scale between 0 and 1 and are interpreted as probabilities in a well-defined random experiment. One of the contributions of the paper is the possibility for integrating recommendations into the model and for reasoning about trust.

The paper draws its motivation from various sources, including Phil Zimmermann's Pretty Good Privacy (PGP) software [27] and previous work on public-key management [26],[1],[3],[19],[6],[18],[2],[25].

There seems to be an inherent trade-off between the efficiency of an implementation of a model on one hand, and the expressive power of the model and the precision of the semantics on the other hand. One particular problem encountered in the literature is that ad-hoc rules for calculating a confidence value from other confidence values are based on probability-theoretic arguments, despite the fact that no random experiment can be specified in which these probabilities are well-defined.

The emphasis of this paper is on precision rather than efficiency. In order to be used efficiently in a large PKI the model might have to be simplified accordingly. A second restriction of our model is that certificate revocation is not yet included. The solution of both these problems is the subject of future research.

The paper is organised as follows. In Section 2 we discuss various aspects of public-key certification, trust, recommendations and the problems involved in defining a model for a public-key infrastructure. Section 3 presents a deterministic model without confidence values, and this model is extended in Section 4 to a probabilistic model incorporating confidence values which are interpreted as probabilities of events in a random experiment. In Section 5 a few open problems and directions for future research are mentioned.

2 Preliminaries

2.1 Cryptography

Cryptographic techniques (e.g., see [21]), in particular public-key cryptosystems and digital signature schemes [7],[20], are of fundamental importance in distributed systems security and electronic commerce. Two typical applications are key management (e.g., the generation of a secret key shared by two entities not sharing a secret key initially, which can be used to set up a secure connection between them) and the generation and verification of digital signatures, for instance on a digital contract, a purchase order, or an email message.

One of the major advantages of public-key cryptographic techniques, compared to conventional cryptographic techniques, is their asymmetry: while only an entity knowing an appropriate secret key can perform a certain operation (e.g. decrypt or sign a message), everyone knowing the corresponding public key can perform a corresponding operation (e.g. encrypt a message or verify a signature).

A public key of an entity or user Bob[2] is completely useless for a user (say Alice) unless she can convince herself that it is authentic, i.e., that it was indeed generated by Bob and therefore that only he knows the corresponding secret key. One of the major problems in public-key management is therefore to provide mechanisms allowing an entity to obtain or retrieve another entity's public key together with evidence of its authenticity.

The authenticity of a public key can either be verified by invoking a non-cryptographic authentication mechanism, for instance by exchanging a hash value of the public key over the phone (assuming that the speaker can be identified on the phone), or by using public-key certificates described in the following section.

2.2 Public-key certification

A public-key certificate is a digital signature, issued by an entity or authority, for a message stating that a certain public key belongs to a certain entity[3]. Alice can use a certificate issued by an entity X for user Bob if and only if the following two conditions are satisfied:

1. Alice knows the public key of X (for verifying the certificate) and is convinced of its *authenticity*.
2. Alice *trusts* X to be honest and to correctly authenticate the owner of a public key before signing it.

If Alice does not know an authentic copy of X's public key, the first condition can be satisfied by using a certificate for X's public key issued by another entity Y. This process can be iterated, thus making use of a chain of certificates. However, Alice can use such a chain of certificates if and only if she trusts every entity in the chain between her and Bob [14].[4]

Public-key certification can be organised in a number of different ways. Among the proposed structures are hierarchical or semi-hierarchical ones (e.g. CCITT X.509 [28], Privacy Enhanced Mail [29]) and distributed approaches as suggested by Phil Zimmermann [27].[5] Various types of certification structures are emerging independently. They will coexist and together form a global PKI.

[2] In this paper we will most often refer to the users Alice and Bob, but the reader should keep in mind that they need not be persons. Their role could be played by an arbitrary entity, for example a server, an application programs, an IP-layer encryption mechanism, a trusted component of an operating system, or a personal token like a chipcard.

[3] A certificate generally contains further information, for instance the date of signing, the expiration date, or the application context for which it is valid.

[4] In PGP, such intermediate entities trusted by Alice are called introducers.

[5] We refer to [23] for a discussion of public-key certification in the context of network security.

Government organisations will install a mostly hierarchical infrastructure as a service to the society, large organisations will typically build their own hierarchical infrastructure within the organisation, business communities (e.g. the banking world) can build a structure for use between organisations of that community, and individual people may become part of a global web according to Zimmermann's "grass roots" approach in which each person takes a share of the responsibility. A given user or system can retrieve, use and combine certificates from arbitrary substructures.

One can expect the growth of a global PKI in the near future, in which arbitrary entities can issue certificates, resulting in a web of certificates that can be represented as a directed graph whose vertices are the entities and where an edge from X to Y means that X has certified Y's public key. It can be expected that some if not most public keys will be certified by several certification authorities and/or users, hence allowing users to select the certificate(s) most suited for their purpose. In consequence, such a web of certificates is likely to be a very large and highly distributed information system.

2.3 Trust and recommendations

Propagating authenticity of public keys by certificates is quite straight-forward. In contrast, it is less obvious how trust should be established and propagated, i.e., how a "web of trust" should be created.[6]

In most previously proposed approaches, including PGP, the propagation of trust is not considered within the model. In PGP, for example, a user can specify which users he or she trusts but the system does not derive any conclusions about the trustworthiness of entities. Such decisions are left completely to the users and are hence dealt with outside of the model.

Recommendations are of fundamental importance in our society because it is impossible to know personally all the people one has to rely on. Such recommendations can be implicit or explicit. The fact that one generally trusts a policeman is an example of an implicit recommendation while a letter of recommendation for a job application is explicit. Yahalom et al. [26] have proposed a public-key management model which includes explicit recommendations. This model was extended in [1]. The model proposed in this paper extends the previous work on explicit recommendations.

A recommendation can be thought of as a signed statement about the trustworthiness of another entity and is similar to a certificate. In contrast to certificates, recommendations can be sensitive information and should sometimes be treated confidentially. This is one of the reasons why PGP does not make use of recommendations. However, confidentiality of recommendations can be implemented by proper encryption and access control mechanisms and is not considered further in this paper.

[6] In the literature, the term "web of trust" is often somewhat misleadingly used to refer to a web of certificates.

Including recommendations in public-key certification does not imply that a user loses control over which recommendations can securely be used in her context. To the contrary, a user can specify precisely a policy according to which recommendation are to be used.

Recommendations are more complicated than certificates. There exist several levels of trust and recommendations in the context of public-key certification. A recommendation of the first level is for someone to be trustworthy for the certification of public keys. A recommendation of the second level is for an entity to be trustworthy in recommending other entities for certification. Generally, a recommendation of the i-th level is for an entity to be trustworthy in giving recommendations of level $i-1$. In a certain sense, a certificate can be interpreted as a special type of recommendation of level 0.

Trust is a resource that fades out very quickly along a path of recommendations. A reasonable system would therefore probably use only a small number of levels of recommendations.

2.4 Confidence valuation and using multiple certification paths

No authentication process is perfect, and nobody is completely trustworthy. As pointed out by Phil Zimmermann in [27], trust in a person can range from marginal to fully trusted, and in fact all intermediate degrees of trust are possible. Similarly, the security of an authentication procedure can range from marginal to fully secure. It is therefore natural to increase the confidence in the authenticity of a public key by verifying several different certificate chains for the same public key (see also [1]). Similarly, several independent recommendations can be combined to obtain a stronger combined recommendation. One way of using confidence parameters is for implementing gradual expiration of certificates, by letting the confidence parameter decrease with time.

In order to be able to combine and exploit several independent certification paths or recommendations it is necessary to measure confidence.[7] It appears natural to use a scale from 0 to 1, where 0 stands for no confidence and 1 stands for complete confidence, and to interpret these values as probabilities. However, defining such a random experiment is non-trivial because all the confidence parameters must be interpreted as probabilities of well-defined events of the *same* random experiment. Otherwise, the meaning of probabilities is undefined. For previously proposed approaches (e.g. [1],[27]) no such random experiment can be defined.

Combining the confidence values of independent parallel certification paths into a higher confidence value for the authenticity of the certified public key could perhaps appear to be quite straight-forward [1]. However, certification graphs are generally more complex because the individual paths intersect. This problem is addressed in Section 4 where the probabilistic model is introduced.

[7] For example, PGP allows the assignment of a confidence parameter to the trustworthiness of an introducer, but it does not consider confidence parameters for the authenticity of public keys. The scale for measuring trust contains four possible values: unknown, marginally trusted, fully trusted, and ultimately trusted.

2.5 Dependencies between parameters

One of the major problems in reasoning with uncertain information are dependencies between different pieces of input information. There are two types of dependencies to be considered in a PKI. Structural dependencies were mentioned in the previous section. For example, if two different certification paths contain the same certificate, then they are obviously not independent. A PKI model must take into account that when this certificate is false for some reason, then both certification paths fail simultaneously.

The second type of dependency are correlations between entities and is more difficult to capture in a model. For example, if two entities belong to the same organisation, their trustworthiness may not be independent. In consequence, two disjoint certification paths each containing one of these entities are not independent. One of the major problems with modelling such dependencies is that, in its most general form, the size of a specification of dependencies is exponential in the number of entities. Therefore every scenario considered in practice is bound to be a special case of some type. Nevertheless, our model allows in principle to take into account arbitrary dependencies.

2.6 Security policies

One can distinguish between (at least) two types of security policies that can be involved in distributed system security: (1) policies that specify how entities and organisations should behave when participating in the development of the PKI and (2) the individual users' policies used for deriving conclusions from the available information.

Several policies of the first type can coexist. Such a policy could specify how confidence parameters should be assigned to certificates and recommendations. For example, it could state that authentication based on speaker identification on a telephone line should be assigned a confidence parameter of at most 0.95 whereas authentication based on the verification of a passport could be assigned an arbitrary confidence level.

A user's security policy (second type) could specify the required confidence levels for certain actions and could specify a maximal confidence level (e.g. 0.9) to be used with recommendations. For example, Alice might be satisfied with a confidence value of 0.3 for verifying Bob's invitation to his birthday party, but she would probably require a very high confidence value for the authenticity of the public key she uses for checking the signature on an important digital contract.

2.7 Requirements for a model of a public-key infrastructure

Three goals of defining a model of a public-key infrastructure are:

- to provide a framework (syntax) for expressing statements and security policies.

- to give precise meaning (semantics) to parameters.
- to provide rules and procedures for analysing a particular scenario and for deriving conclusions.

Some of the requirements for such a model are listed below.

- *Generality and expressive power.* The model should capture all aspects of public-key certification, including trust, recommendations, confidence values for trust and authenticity of public keys, multiple certification paths, the revocation of public keys, and dependencies between parameters.
- *Precise Semantics.* The parameters of the model should have a clear interpretation. In particular, when probabilities are used, it should be possible to interpret all confidence values as probabilities of events in a single (overall) random experiment.
- *Evaluation order independence.* The derived conclusions should be independent of the order in which rules are applied or, at least, the order of applying rules should be uniquely specified. Certification or recommendation cycles should not lead to instable feedback in the application of evaluation rules.
- *Efficient implementation.* The model should be suitable for an efficient implementation, i.e., the algorithms for deriving conclusions should be efficient.
- *Scalability.* It should be possible to treat entity populations of arbitrary size, to easily update the parameters when new entities are included in a view, and to implement policies of significant complexity.
- *Easy usability.* The specification of the parameters should be intuitive and the model should be easy to work with.

Clearly, some of these requirements are conflicting. In particular, expressive power and generality are in conflict with efficient implementation and easy usability. It appears impossible to satisfy both types of requirements perfectly, and the focus of Section 4 of this paper is biased towards generality and expressive power. Any completely general model will probably have to be simplified to be used in practice, but such a simplification should be made in full awareness of the restrictions it implies on the general model.

3 A deterministic model for public-key certification, trust and recommendations

We briefly sketch the basic ideas behind our deterministic model that is based on a special type of logic. The syntax is very simple: the propositions or formulas (referred to as *statements* in our context) are simple expressions that take one of four different forms (see Definition 3.1). For example, $Aut_{A,B}$ denotes the statement that, from A(lice)'s point of view, her copy of Bob's public key is authentic. The syntax contains no Boolean operators ($\wedge$, $\vee$, $\neg$) or quantifiers ($\exists$, $\forall$).

The semantics is based on two inference rules (for authenticity and trust) for deriving statements from sets of statements. The axioms are a set of statements (certificates, recommendations and initial authenticity and trust assignments) considered true by Alice, and the set of axioms is called Alice's initial view. In contrast to classical propositional logic [16], the truth values assigned to statements are not true and false, but *valid* and *invalid.* A statement is valid (in Alice's view) if and only if it can be derived from the axioms (her initial view). An invalid statement is not necessarily false (in a normal sense), but if it is true, then Alice has no evidence of this fact. Alice's derived view is the set of statements derivable from the axioms.

Before defining the model more formally, let us review Alice's procedure for establishing the authenticity of (say) Bob's public key, i.e., the validity of the statement $Aut_{A,B}$. Alice builds her initial view (the set of axioms) by collecting statements that can be relevant in the context of authenticating Bob's public key. There are two categories of statements, namely those provided by other entities (by making them accessible through the PKI) and retrieved by Alice from the PKI[8] (certificates and recommendations), and those specified by Alice as part of her belief (authenticity of certain public keys, trust in certain entities). Each of these categories consists of two types of statements, one referring to the authenticity of public keys and one referring to the trustworthiness of entities, resulting in a total number of four types of statements.

Such statements will in figures be depicted as edges (solid or dashed) in a directed graph in which the vertices correspond to entities. The graph represents the web of certificates, trust and recommendations available to Alice. Authenticity and certificates are represented by solid edges and trust and recommendations by dashed edges. There are several levels of trust and recommendations (as explained in Section 2.3 and below), and dashed edges are labelled with the corresponding level. This is summarised in Definitions 3.1 and 3.2.

Definition 3.1. *Statements* are of one of the following forms:

- *Authenticity of public keys.* $Aut_{A,X}$ denotes Alice's belief that a particular public key P_X is authentic (i.e., belongs to entity X) and is represented graphically as an edge from A to X: $A \longrightarrow X$.
- *Trust.* $Trust_{A,X,1}$ denotes Alice's belief that a particular entity X is trustworthy for issuing certificates. Similarly, her belief that X is trustworthy for issuing recommendations of level $i-1$ is denoted by $Trust_{A,X,i}$. The symbol is a dashed edge from A to X labelled with the trust level: $A \overset{i}{\dashrightarrow} X$.
- *Certificates.* $Cert_{X,Y}$ denotes the fact that Alice holds a certificate for Y's public key (allegedly)[9] issued and signed by entity X. The symbol is an edge from X to Y: $X \longrightarrow Y$.
- *Recommendations.* $Rec_{X,Y,i}$ denotes the fact that Alice holds a recommendation of level i for entity Y (allegedly) issued and signed by entity X. The symbol is a dashed edge from X to Y labelled with i: $X \overset{i}{\dashrightarrow} Y$.

[8] By "retrieving from the PKI" we mean any method of obtaining certificates or recommendations, for instance by accessing a certificate server or by asking the owner of a certificate to provide it.

[9] We use the word "alleged" because without verification, there exists no evidence that the certificate was indeed issued by the claimed entity.

Alice's *initial view*, denoted $View_A$, is a set of statements.

As the symbols suggest, authenticity could be interpreted as a special type of certification (i.e., signed by Alice's own secret key which is ultimately trusted). Similarly, trust could be interpreted as a special type of recommendation signed by Alice (see also remark 3 at the end of this section). We will not use this simplified notation.

Let us now describe the inference rules of our model, i.e., what it means to derive statements from other statements. The conclusions derived by Alice within the model are statements of one of the first two types of Definition 3.1. In all our examples, the ultimate goal will be to derive the statement $Aut_{A,B}$, namely that in Alice's view her copy of Bob's public key is authentic.

Definition 3.2. A statement is *valid* if and only if it is either contained in $View_A$ or if it can be derived from $View_A$ by applications of the following two inference rules:

$$\forall X,Y: \quad Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Y}\ \vdash\ Aut_{A,Y} \tag{1}$$

and

$$\forall X,Y,i\geq 1: \quad Aut_{A,X},\ Trust_{A,X,i+1},\ Rec_{X,Y,i}\ \vdash\ Trust_{A,Y,i}\,. \tag{2}$$

For a finite set $\mathcal{S}$ of statements, $\overline{\mathcal{S}}$ denotes the closure of $\mathcal{S}$ under applications of the inference rules (1) and (2), i.e., the set of statements derivable from $\mathcal{S}$. Alice's *derived view* is the set $\overline{View_A}$ of statement derivable from her initial view $View_A$. A statement S is hence valid if and only if $S \in \overline{View_A}$, and invalid otherwise.

The first rule is for deriving statements about the authenticity of public keys. It states that Alice can derive the authenticity of a certified public key for user Y (denoted $Aut_{A,Y}$) if for some entity X who has certified Y's public key (denoted $Cert_{X,Y}$) she can derive the authenticity of X's public key (denoted $Aut_{A,X}$) and trust of level 1 into entity X (denoted $Trust_{A,X,1}$).

The second rule is for deriving statements about trust. It states that for all $i\geq 1$, if Alice has trust of level $i+1$ in X (denoted $Trust_{A,X,i+1}$) then she accepts a recommendation from X of level i for another entity Y (denoted $Rec_{X,Y,i}$), provided that she believes that her copy of X's public key is authentic (denoted $Aut_{A,X}$).

We will assume throughout the paper that trust and recommendations of level i imply trust and recommendations of lower levels, i.e.,

$$\forall X,Y,1\leq k<i: \quad Trust_{A,X,i}\ \vdash\ Trust_{A,X,k} \tag{3}$$

and

$$\forall X,Y,1\leq k<i: \quad Rec_{X,Y,i}\ \vdash\ Rec_{X,Y,k}\,. \tag{4}$$

These rules, which are not part of the model, appear to be intuitive, but they are not essential for the model. For instance, it seems to make no sense to specify trust of levels 1 and 3 (but not 2) in a certain entity.

Note that an initial view $View_A$ need not necessarily be minimal in the sense that a statement S cannot be derived from the remaining set of statements, $View_A - \{S\}$. We now explain the model by a number of simple examples.

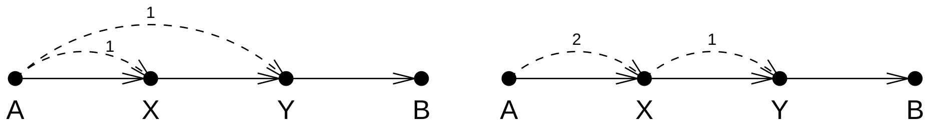

Figure 1.

Example 3.3. Consider a chain of certificates shown in Figure 1 (left). Alice has specified her copy of X's public key as authentic ($Aut_{A,X}$). Her view also contains two certificates: a certificate for Y (allegedly) issued by X ($Cert_{X,Y}$) and a certificate for B allegedly issued by Y ($Cert_{Y,B}$). Furthermore, Alice trusts both X and Y to correctly certify public keys. More formally, we have

$$View_A = \{Aut_{A,X},\ Cert_{X,Y},\ Cert_{Y,B},\ Trust_{A,X,1},\ Trust_{A,Y,1}\}$$

and the statement $Aut_{A,B}$ can be derived by two applications of rule (1):

$$\begin{aligned} Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Y} &\vdash Aut_{A,Y} \\ Aut_{A,Y},\ Trust_{A,Y,1},\ Cert_{Y,B} &\vdash Aut_{A,B}. \end{aligned}$$

Hence we have

$$\overline{View_A} = View_A \cup \{Aut_{A,Y},\ Aut_{A,B}\}.$$

Example 3.4. The scenario of Figure 1 (right) shows the same chains of certificates, but Alice does not trust Y initially. However, she trusts X of level 2 ($Trust_{A,X,2}$) and is hence willing to accept recommendations of the first level from X. The statement $Aut_{A,B}$ can be derived in a similar way as described in Example 3.3, but the statement $Trust_{A,Y,1}$ is not contained in Alice's view and must therefore be derived:

$$\begin{aligned} Aut_{A,X},\ Trust_{A,X,2},\ Rec_{X,Y,1} &\vdash Trust_{A,Y,1} \\ Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Y} &\vdash Aut_{A,Y} \\ Aut_{A,Y},\ Trust_{A,Y,1},\ Cert_{Y,B} &\vdash Aut_{A,B} \end{aligned}$$

Example 3.5. A slightly more complicated scenario is shown in Figure 2 (left). The derivation of $Aut_{A,B}$ is achieved by the following steps:

$$\begin{aligned} Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Y} &\vdash Aut_{A,Y} \\ Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Z} &\vdash Aut_{A,Z} \end{aligned}$$

$$Aut_{A,W},\ Trust_{A,W,3},\ Rec_{W,Z,2}\ \vdash\ Trust_{A,Z,2}$$
$$Aut_{A,Z},\ Trust_{A,Z,2},\ Rec_{Z,Y,1}\ \vdash\ Trust_{A,Y,1}$$
$$Aut_{A,Y},\ Trust_{A,Y,1},\ Cert_{Y,B}\ \vdash\ Aut_{A,B}$$

In this example we have

$$View_A = \{Aut_{A,X},\ Aut_{A,W},\ Cert_{X,Y},\ Cert_{X,Z},\ Cert_{Y,B},\ Trust_{A,X,1},\ Trust_{A,W,3},\ Rec_{W,Z,2},\ Rec_{Z,Y,1}\}$$

and

$$\overline{View_A} = View_A \cup \{Aut_{A,Y},\ Aut_{A,Z},\ Aut_{A,B},\ Trust_{A,Z,2},\ Trust_{A,Y,1}\}.$$

In general, the derived view $\overline{View_A}$ contains more statements than necessary for deriving $Aut_{A,B}$. Hence it is often unnecessary to determine $\overline{View_A}$ completely.

The previous example illustrates that recommendations and certificates can be issued independently. An entity can even recommend an entity whose public key it does not know. For example, W has issued a recommendation for Z without certifying Z's public key. While issuing a certificate requires an authenticated copy of the public key to be certified, a recommendation can be issued without prior exchange of information with the entity that is being recommended.

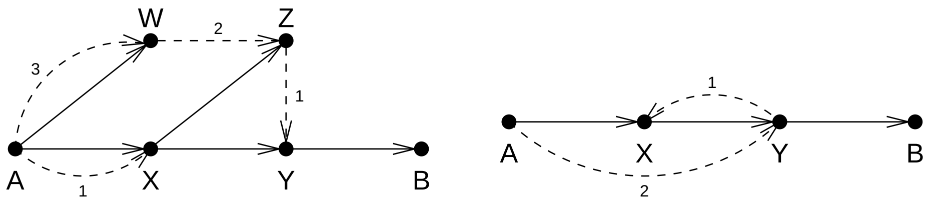

Figure 2.

Example 3.6. Consider the simple scenario of Figure 2 (right). Alice has trust of level 2 in Y and Y has issued a recommendation for X. It may appear that Alice can make use of this recommendation to use the certificate $Cert_{X,Y}$ for deriving $Aut_{A,Y}$. However, the derivation of $Aut_{A,Y}$ should be impossible because if X is dishonest (which is consistent with Alice's view), he can generate fake certificates $Cert_{X,Y}$ and $Cert_{Y,B}$ and a fake recommendation $Rec_{Y,X,1}$ without Y being involved or aware of the attack. The reader can verify that indeed neither the statement $Aut_{A,Y}$ nor the statement $Aut_{A,B}$ can be derived from

$$View_A = \{Aut_{A,X},\ Cert_{X,Y},\ Cert_{Y,B},\ Trust_{A,Y,2},\ Rec_{Y,X,1},\}$$

and in fact we have $\overline{View_A} = View_A$.

In a typical application, if $Aut_{A,B}$ cannot be derived, Alice (or her system) could try to enlarge the initial view $View_A$ by retrieving more certificates and recommendations from the PKI, by specifying further entities as trustworthy, and/or by directly authenticating some of the public keys (e.g. by checking a hashed value of the public key during a telephone call).

In a model without recommendations (e.g. PGP), and hence assuming that (1) is the only inference rule, a result of [3], [14] and [19] can be restated as follows. The proof is straight-forward and is omitted.

Theorem 3.7. *If Alice's initial view ($View_A$) contains no recommendations, then $Aut_{A,B}$ can be derived if and only if there exists a complete chain of certificates from Alice to Bob and Alice trusts every intermediate entity with respect to certification. Formally, we have $Aut_{A,B} \in \overline{View_A}$ if and only if either $Aut_{A,B} \in View_A$ or for some $k \geq 1$ there exist $X_1, \ldots, X_k$ such that $View_A$ contains Aut_{A,X_1}, $Cert_{X_j,X_{j+1}}$ for $j = 1, \ldots, k-1$, $Cert_{X_k,B}$ and $Trust_{A,X_j,1}$ for $j = 1, \ldots, k$.*

Remarks.

1. Entities should not certify public keys whose authenticity was derived within the model (by application of the inference rules). This could lead to undesirable hidden effects for other entities if the individual entities' policies are not known. Assume for example that in Figure 1 (left) entity X's level-1 recommendation for Y ($Rec_{X,Y,1}$) is a consequence of a higher-level recommendation but that Alice's policy is to discard any recommendations of level greater than 1. If Alice knew that X used a high-level recommendation for deriving $Rec_{X,Y,1}$, Alice would have to discard this statement.
2. The above notation could be simplified by the syntactic conventions that authenticity is equivalent to trust of level 0 and that the certification of a public key corresponds to a recommendation of level 0. Together with the simplification mentioned earlier, one could summarise the four types of statements into one category of statements denoted by $S_{X,Y,i}$, where $S_{A,X,0}$, $S_{A,X,i}$, $S_{X,Y,0}$ and $S_{X,Y,i}$ stand for $Aut_{A,X}$, $Trust_{A,X,i}$, $Cert_{X,Y}$ and $Rec_{X,Y,i}$, respectively, for all $i \geq 1$. For such a simplified notation the two rules (1) and (2) can be summarised in a single inference rule:

$$\forall X, Y, i \geq 0: \quad S_{A,X,0},\ S_{A,X,i+1},\ S_{X,Y,i} \ \vdash\ S_{A,Y,i}$$

3. A number of approaches based on logic for reasoning about security have previously been proposed [4],[10],[24],[5], but they are not directly applicable in our context. Other papers describing calculi of trust and authenticity are [3],[18],[19] and [14], but they do not consider recommendations nor confidence values.

4 Confidence valuation: a model based on probabilistic logic

In a realistic scenario, the statements used in the derivation of a certain conclusion are never completely certain. Trust in a person can vary from marginal to complete, and the authenticity of a given public key may depend on the method used for checking the authenticity. It appears natural to measure the confidence in the validity of a statement on a continuous scale between 0 and 1 and, if possible, to interpret the value in some sense as the probability that the statement is correct. The goal of this section is to present a formal model in which the confidence values of all statements can be interpreted as probabilities in a well-defined random experiment (or probability structure).

One can distinguish different approaches to integrating probabilities into a deterministic model based on inference rules.

- Perhaps a natural approach (see for instance [1]) appears to be to incorporate confidence values into the inference rules, i.e., to use rules that specify the degree of confidence of a conclusion as a function of the confidence values of the preconditions of the rule. However, in such an approach the confidence values of derivable statements will generally depend on the order in which the rules are applied. Moreover, certification and recommendation cycles can lead to an undesirable amplification of confidence, and it appears impossible to model dependencies between the confidence values for different statements. Furthermore, it appears very difficult (if not impossible) to describe the meaning of such derivations, i.e. to describe a random experiment in which the confidence values can be interpreted as probabilities of naturally defined events.
- The second approach, which we have chosen for this paper, preserves the convenience of deterministic inference rules, but considers the initial view to be uncertain (i.e., a random variable). More precisely, we assume a probability distribution over the possible initial views, and the confidence value of a statement is defined as the probability that it can be derived from the initial view.

4.1 Reasoning with uncertain information

Reasoning with uncertain information is an important research area in artificial intelligence. Several approaches and models have been proposed and some of them have been used in the implementation of expert systems. We discuss very briefly some of the issues that have been considered and refer to [8], [12] and [17] for further references to the literature.

A view often taken in probability theoretic approaches to reasoning with uncertain information is that the sample space of the probability structure is a set of *possible worlds* and that the real world corresponds to one of these worlds, each with a certain probability. The probability that a statement is true is the total probability of all worlds in which the statement is true. Possible worlds differ in the axioms, and they can even differ in the set of inference rules that are applicable in the world (e.g., see [5]). A natural requirement is that when all probabilities are either 0 or 1, then the model coincides with a natural deterministic model. As in most papers on probabilistic logic, we assume that the inference rules (of Definition 3.2) are universally applicable in all worlds but that the set of statements assumed by Alice to be valid (i.e., the axioms contained in her initial view) are different in different worlds. Probabilistic logic has previously been applied in the analysis of security protocols in [5].

One potential problem with the described approach is that not all sets of worlds need be measurable according to the probability measure[10]. Several researchers have investigated solutions to this problem, and Fagin and Halpern have even intentionally defined structures containing non-measurable sets.

A very interesting problem (e.g., see [8]), which is not considered in this paper, is for given specified probabilities of statements to compute the admissible intervals of probabilities for the other statements, i.e., to determine the extreme values of these probabilities that are attainable for a probability measure consistent with the given probabilities. A term sometimes used in this context is belief functions.

4.2 The probabilistic model of a PKI

In the probabilistic model (see definition below) we replace the view in the deterministic model by a probability distribution over a finite set of possible views, i.e., by a random variable taking as values deterministic views. Note that for a finite sample space S, the events are all the subsets of S, and a probability measure is specified completely by assigning probabilities to all the sample points. The probability of an event is the sum of the probabilities of the sample points it contains.

Definition 4.1. Let $\mathcal{S}_A$ be the set of statements (of the forms $Aut_{A,\cdot}$, $Trust_{A,\cdot,\cdot}$, $Cert_{\cdot,\cdot}$ and $Rec_{\cdot,\cdot,\cdot}$) that Alice considers as possible elements of her initial view[11]. The sample space of the random experiment (i.e., the set of possible worlds considered by Alice) is the power set[12] of $\mathcal{S}_A$, denoted by $2^{\mathcal{S}_A}$. Alice's *probabilistic initial view* is a pair $[\mathcal{S}_A, P]$ where $P : 2^{\mathcal{S}_A} \to \mathbb{R}^+$ is a probability function on the sample space $2^{\mathcal{S}_A}$, which naturally extends to a probability measure for all

[10] A probability structure is a triple $(S, \mathcal{X}, P)$ consisting of the sample space S, a σ-algebra $\mathcal{X}$ of subsets of S (i.e., a set of subsets of S containing S and closed under complementation and countable union, but not necessarily consisting of all subsets of S), and a probability measure P assigning a non-negative probability to every element of $\mathcal{X}$ such that the probability of a union of disjoint sets is the sum of their probabilities and $P(S) = 1$.

[11] In the probabilistic model, $S \in \mathcal{S}_A$ does not necessarily imply that the statement S is valid, but rather that it is valid with some probability. In our examples, $\mathcal{S}_A$ consists of the statements represented by edges in the figures.

[12] the set of subsets

events (i.e., sets of subsets of $\mathcal{S}_A$)[13]. Alice's view $View_A$ now denotes the random variable associated with P, i.e., the random variable that takes on as values the subsets of $\mathcal{S}_A$ with the corresponding probabilities.

Note that this definition of $View_A$ is consistent with that given in Section 3 if one assigns probability 1 to one particular initial view. The events in our random experiment are the subsets of the sample space, i.e., the $2^{2^{|\mathcal{S}_A|}}$ sets of subsets of $\mathcal{S}_A$. Let $\mathcal{V}$ be a subset of $\mathcal{S}_A$. Then $P(\mathcal{V})$ and $P(View_A = \mathcal{V})$ denote the same probability, namely the probability of the elementary event (sample point) $\mathcal{V}$. The event $\mathcal{V} \subseteq View_A$ (which is an abbreviation of $\{\mathcal{U} \subseteq \mathcal{S}_A : \mathcal{V} \subseteq \mathcal{U}\}$), whose probability is denoted by $P(\mathcal{V} \subseteq View_A)$, consists of those subsets of $\mathcal{S}_A$ that contain $\mathcal{V}$.

Because $View_A$ is a random variable taking as values sets of statements, so is $\overline{View_A}$, which takes on as values subsets of $\overline{\mathcal{S}_A}$. For a given statement S in $\mathcal{S}_A$ or derivable from $\mathcal{S}_A$, we can consider the event that S can be derived from $View_A$, i.e., the event $S \in \overline{View_A}$. This event consists of those subsets of $\mathcal{S}_A$ (i.e. of those elementary events) from which S can be derived, and its probability is hence the sum of the probabilities of these subsets. Statements not derivable from $\mathcal{S}_A$ correspond to the empty event (probability 0). Later we will characterize the event $S \in \overline{View_A}$ by the union of events of the form $\mathcal{V}_i \subseteq View_A$, where the $\mathcal{V}_i$ are subsets of $\mathcal{S}_A$ from which S can be derived. This is summarised in the following definition.

Definition 4.2. The *confidence value* of a statement $S \in \overline{\mathcal{S}_A}$, denoted $conf(S)$, is the probability that it can be derived from $\mathcal{S}_A$, i.e., it is

$$conf(S) \;=\; P(S \in \overline{View_A}) \;=\; \sum_{\mathcal{V} \subseteq \mathcal{S}_A:\, S \in \overline{\mathcal{V}}} P(\mathcal{V}).$$

This model allows to specify arbitrary dependencies between the statements in $\mathcal{S}_A$, by specifying an appropriate probability measure P. For example, the trustworthiness of two entities X and Y belonging to the same organisation could be modelled to be correlated[14].

While such dependencies can be specified by an appropriate choice of P, it is important to notice that dependencies due to intersecting certification paths are captured by the model itself, as are the dependencies due to the fact that certificates and/or recommendations issued by a single entity, if they fail, are likely to fail simultaneously. In fact, capturing such dependencies is one of the major purposes of introducing our model.

[13] In the following we will use P to denote both the probability function (with $2^{|\mathcal{S}_A|}$ arguments) as well as the implied probability measure (which formally is a function with $2^{2^{|\mathcal{S}_A|}}$ arguments), and we will use the term probability measure for both.

[14] If they are perfectly correlated, this would imply that the probability measure P is 0 for all subsets of $\mathcal{S}_A$ containing $Trust_{A,X,1}$ but not $Trust_{A,Y,1}$ (or vice versa).

4.3 Independent initial confidence parameters

The probability measure P can be specified by specifying its value for the $2^{|\mathcal{S}_A|}$ sample points. The probability of an event is the sum of the probabilities of the corresponding sample points. We will therefore later only specify P for all sample points (subsets of $\mathcal{S}_A$.

In its most general form, a probabilistic view can be intractably complex, and in a realistic scenario one can only consider measures P that can be specified by a reasonable number of parameters. Note that this potential complexity is inherent to the problem and is not due to the choice of our model.

An interesting and often natural restriction for the measure P is to assume that the confidence parameters initially assigned to all the statements in Alice's initial view $\mathcal{S}_A$ (i.e., the edges in the graph) are independent. Let $p(S)$ be the confidence parameter assigned initially by Alice to the statement $S \in \mathcal{S}_A$.

When $\mathcal{S}_A$ is minimal in the sense that no statement $S \in \mathcal{S}_A$ can be derived from the remaining set of statements, $\mathcal{S}_A - \{S\}$, then we have $conf(S) = p(S)$ for all S. However, $\mathcal{S}_A$ is not minimal in general. For instance, in example 4.6 (Figure 4, right), the statement $Aut_{A,Y}$ is in $\mathcal{S}_A$ but it can also be derived from the statements $Aut_{A,X}$, $Cert_{X,Y}$ and $Trust_{A,X,1}$. For statements S that can be derived from $\mathcal{S}_A - \{S\}$, we generally have $conf(S) > p(S)$: $conf(S)$ is the sum of $p(S)$ and the total probability of all subsets of $\mathcal{S}_A - \{S\}$ from which S can be derived. (This is why we refer to $p(S)$ as the initial confidence parameter and to $conf(S)$ as the confidence value.)

It should be mentioned again that our model does not require an independence assumption, but for the sake of simplicity all the examples considered below are based on it. However, there is one exception: according to (3) and (4) we assume that trust and recommendations of levels higher than 1 imply trust and recommendations, respectively, of all lower levels. (The only example involving recommendations, and hence to which the previous comment applies, is example 4.8.)

We are interested in computing the probability that particular subsets, usually minimal[15] subsets from which $Aut_{A,B}$ can be derived, are contained in the initial view $View_A$. When no trust and recommendation statements of levels greater than 1 are in $\mathcal{S}_A$, then the probability that a subset $\mathcal{V}$ of $\mathcal{S}_A$ is a subset of Alice's initial view $View_A$ is the product of the $p(S)$, where S ranges over the set $\mathcal{V}$. In order to take into account the rules (3) and (4), we must consider only the highest level of trust and recommendation statements (for fixed entities involved) and delete those implied by (3) and (4). Let $\mathcal{V}^*$ be the resulting reduced set of statement. Hence we have

$$P(\mathcal{V} \subseteq View_A) = \prod_{S \in \mathcal{V}^*} p(S) \tag{5}$$

if $\mathcal{V}$ is consistent with (3) and (4), and $P(\mathcal{V} \subseteq View_A) = 0$ otherwise. Note again

[15] In the following, *minimal* means that when one statement is deleted from the set, then $Aut_{A,B}$ cannot be derived.

that when no trust and recommendation statements of levels greater than 1 are in $\mathcal{S}_A$, then $\mathcal{V}^* = \mathcal{V}$.

The reader should not be confused by the fact that $\mathcal{V} \subset \mathcal{V}'$ implies $P(\mathcal{V} \subseteq \mathit{View}_A) \geq P(\mathcal{V}' \subseteq \mathit{View}_A)$. Note that $\mathcal{S}_A$ is *not* the sample space, and neither $\mathcal{S}_A$ nor the empty set correspond to the certain event. Although this will generally not be needed, we describe how the probability of an elementary event, $P(\mathcal{V})$, can be computed for the case where there are no trust and recommendation statements of levels greater than 1:

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

In the general case, $P(\mathcal{V})$ is defined similarly.

4.4 Implementation aspects and examples

The figures corresponding to the examples should be interpreted as follows: $\mathcal{S}_A$ consists of the statements represented by the edges (dashed or solid) in the graph. Every edge is labelled with the probability $p(S)$ that Alice assigns initially to the statement S represented by the edge. Remember that $\mathit{conf}(S) > p(S)$ is possible.

Consider the problem of computing the confidence value for the statement $\mathit{Aut}_{A,B}$ (or for any other statement derivable from Alice view). Applying the formula given in Definition 4.2 would require the explicit computation of all the subsets of $\mathcal{S}_A$ from which $\mathit{Aut}_{A,B}$ can be derived, and adding up their probabilities. This requires an exponential number of steps.

A generally much more efficient algorithm is obtained by determining all the minimal subsets, $\mathcal{V}_1, \ldots, \mathcal{V}_k$, from which $\mathit{Aut}_{A,B}$ can be derived. They correspond to certification paths from A to B together with the corresponding trust verification statements (which can of course also contain other certification paths). The event that $\mathit{Aut}_{A,B}$ can be derived consists of all the subsets of $\mathcal{S}_A$ containing at least one of these minimal sets, i.e., it is the union of the events $\mathcal{V}_i \subseteq \mathit{View}_A$ for $i = 1, \ldots, k$:

$$\mathit{conf}(\mathit{Aut}_{A,B}) = P(\bigvee_{i-1}^{k} (\mathcal{V}_i \subseteq \mathit{View}_A)).$$

According to the inclusion-exclusion principle, the probability of the union of k events can be computed by taking the sum of their probabilities, subtracting the probabilities of all $\binom{k}{2}$ events resulting from intersecting[16] 2 events, adding

[16] It may at first appear counter-intuitive that the intersection of the two events $\mathcal{V}_i \subseteq \mathit{View}_A$ and $\mathcal{V}_j \subseteq \mathit{View}_A$ is the event $(\mathcal{V}_i \cup \mathcal{V}_j) \subseteq \mathit{View}_A$, involving the union of the two sets $\mathcal{V}_i$ and $\mathcal{V}_j$. However, understanding this fact is a key to understanding our probabilistic model.

the probabilities of all $\binom{k}{3}$ events resulting from intersecting 3 events, etc. This gives

$$\begin{aligned} conf(Aut_{A,B}) = & \sum_{i=1}^{k} P(\mathcal{V}_i \subseteq View_A) \\ & - \sum_{1 \le i_1 < i_2 \le k} P((\mathcal{V}_{i_1} \cup \mathcal{V}_{i_2}) \subseteq View_A) \\ & + \sum_{1 \le i_1 < i_2 < i_3 \le k} P((\mathcal{V}_{i_1} \cup \mathcal{V}_{i_2} \cup \mathcal{V}_{i_3}) \subseteq View_A) \\ & - \cdots \end{aligned}$$

The complexity of computing the confidence value for a statement is on the order of $2^{\min(|\mathcal{S}_A|,k)}$, where k is the number of minimal subsets of $\mathcal{S}_A$ from which the statement can be derived.

The numerical values we have chosen in the following examples are probably smaller than what they would be in a real-life example, but they illustrate better the effect of parallel certification paths, recommendations, etc.

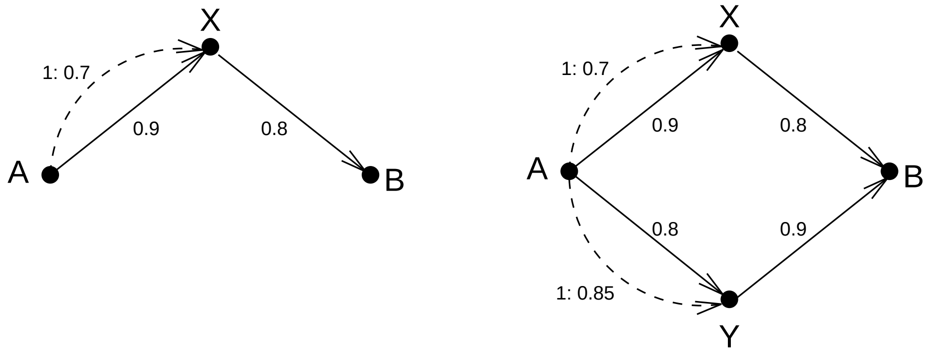

Figure 3.

Example 4.3. A simple example is shown in Figure 3 (left). Alice has assigned confidence parameters 0.9 and 0.7 to the statements $Aut_{A,X}$ and $Trust_{A,X,1}$, respectively. Entity X has assigned confidence parameter 0.8 to the authenticity of B's public key and this confidence parameter is included in the certificate $Cert_{X,B}$. The statement $Aut_{A,B}$ can be derived from

$$\mathcal{S}_A = \{Aut_{A,X}, Trust_{A,X,1}, Cert_{X,B}\}$$

but from no proper subset of $\mathcal{S}_A$. Therefore its confidence value is the product of the three confidence parameters:

$$\begin{aligned} conf(Aut_{A,B}) &= P(\mathcal{S}_A \subseteq View_A) = P(View_A = \mathcal{S}_A) \\ &= p(Aut_{A,X}) \cdot p(Trust_{A,X,1}) \cdot p(Cert_{X,B}) \\ &= 0.9 \cdot 0.7 \cdot 0.8 = 0.504. \end{aligned}$$

Example 4.4. A slightly more complicated example is shown in Figure 3 (right). We have

$$\mathcal{S}_A = \{Aut_{A,X},\ Aut_{A,Y},\ Trust_{A,X,1},\ Trust_{A,Y,1},\ Cert_{X,B},\ Cert_{Y,B}\}$$

and the statement $Aut_{A,B}$ can be derived from any subset of $\mathcal{S}_A$ containing either the minimal set

$$\mathcal{V}_1 = \{Aut_{A,X}, Trust_{A,X,1}, Cert_{X,B}\}$$

or the minimal set

$$\mathcal{V}_2 = \{Aut_{A,Y}, Trust_{A,Y,1}, Cert_{Y,B}\}.$$

We have

$$\begin{aligned} P(\mathcal{V}_1 \subseteq View_A) &= p(Aut_{A,X}) \cdot p(Trust_{A,X,1}) \cdot p(Cert_{X,B}) \\ &= 0.9 \cdot 0.7 \cdot 0.8 = 0.504 \end{aligned}$$

and

$$\begin{aligned} P(\mathcal{V}_2 \subseteq View_A) &= p(Aut_{A,Y}) \cdot p(Trust_{A,Y,1}) \cdot p(Cert_{Y,B}) \\ &= 0.8 \cdot 0.85 \cdot 0.9 = 0.612. \end{aligned}$$

Because the sets $\mathcal{V}_1$ and $\mathcal{V}_2$ are disjoint we have

$$P((\mathcal{V}_1 \cup \mathcal{V}_2) \subseteq View_A) = P(\mathcal{V}_1 \subseteq View_A) \cdot P(\mathcal{V}_2 \subseteq View_A)$$

and hence

$$\begin{aligned} conf(Aut_{A,B}) &= P((\mathcal{V}_1 \subseteq View_A) \vee (\mathcal{V}_2 \subseteq View_A)) \\ &= P(\mathcal{V}_1 \subseteq View_A) + P(\mathcal{V}_2 \subseteq View_A) - P((\mathcal{V}_1 \cup \mathcal{V}_2) \subseteq View_A) \\ &= 0.504 + 0.612 - 0.504 \cdot 0.612 = 0.8076. \end{aligned}$$

Example 4.5. The previous example could be treated intuitively (cf. [1]) by the simple observation that the two paths $A-X-B$ and $A-Y-B$ are independent. Consider now the situation of Figure 4 (left), which is obtained from the previous example by replacing B by a new entity Z who has certified B's public key and is trusted by Alice. In other words, we have

$$\begin{aligned} \mathcal{S}_A = \{&Aut_{A,X}, Aut_{A,Y}, Trust_{A,X,1}, Trust_{A,Y,1}, Trust_{A,Z,1}, Cert_{X,Z}, \\ &Cert_{Y,Z}, Cert_{Z,B}\}. \end{aligned}$$

In this example, the two paths $A-X-Z-B$ and $A-Y-Z-B$ are not independent. Nevertheless, the computation of $conf(Aut_{A,B})$ is intuitive. $Aut_{A,B}$ can be derived from $View_A$ if it contains $Trust_{A,Z,1}$ and $Cert_{Z,B}$ and if $Aut_{A,Z}$ can be derived. The latter condition is equivalent to the condition that $Aut_{A,B}$ can be derived in the previous example. $conf(Aut_{A,Z})$ is hence equal to $conf(Aut_{A,B})$ in the previous example and we have

$$\begin{aligned} conf(Aut_{A,B}) &= conf(Aut_{A,Z}) \cdot p(Trust_{A,Z,)} \cdot p(Cert_{Z,B}) \\ &= 0.8076 \cdot 0.7 \cdot 0.95 = 0.537. \end{aligned}$$

Although there are two certification paths from A to B, the confidence value is low because the two paths intersect in the vertex Z and the edge $Z-B$.

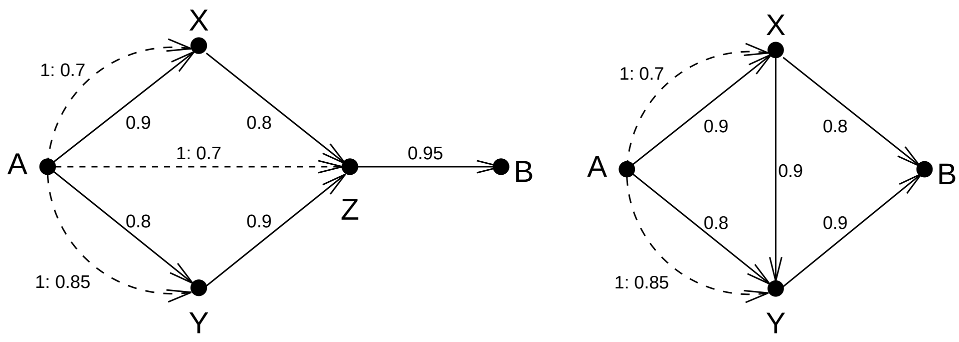

Figure 4.

Example 4.6. The example of Figure 4 (right) is more involved because the three different certification paths $A-Y-B$, $A-X-Y-B$ and $A-X-B$ overlap in a more complicated manner. The calculation of $conf(Aut_{A,B})$ is therefore more complex, and this is the first example that cannot be treated in an intuitive manner. We have

$$\mathcal{S}_A = \{Aut_{A,X},\ Aut_{A,Y},\ Trust_{A,X,1},\ Trust_{A,Y,1},\ Cert_{X,B},\ Cert_{Y,B},\ Cert_{X,Y}\}$$

and the statement $Aut_{A,B}$ can be derived from $View_A$ if and only if it contains as a subset one of the following sets:

$$\begin{aligned} \mathcal{V}_1 &= \{Aut_{A,X}, Trust_{A,X,1}, Cert_{X,B}\}, \\ \mathcal{V}_2 &= \{Aut_{A,Y}, Trust_{A,Y,1}, Cert_{Y,B}\}, \\ \mathcal{V}_3 &= \{Aut_{A,X}, Trust_{A,X,1}, Trust_{A,Y,1}, Cert_{X,Y}, Cert_{Y,B}\}. \end{aligned}$$

Considered as an event in our random experiment, the statement $Aut_{A,B}$ thus corresponds to the set of subsets of $\mathcal{S}_A$ which contain at least one of these sets. The probability of this event can be computed by the exclusion-inclusion principle as

$$\begin{aligned} conf(Aut_{A,B}) &= P((\mathcal{V}_1 \subseteq View_A) \vee (\mathcal{V}_2 \subseteq View_A) \vee (\mathcal{V}_3 \subseteq View_A)) \\ &= P(\mathcal{V}_1 \subseteq View_A) + P(\mathcal{V}_2 \subseteq View_A) + P(\mathcal{V}_3 \subseteq View_A) \\ &\quad -P((\mathcal{V}_1 \cup \mathcal{V}_2) \subseteq View_A) - P((\mathcal{V}_1 \cup \mathcal{V}_3) \subseteq View_A) \\ &\quad -P((\mathcal{V}_2 \cup \mathcal{V}_3) \subseteq View_A) + P((\mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3) \subseteq View_A) \\ &= 0.825 \end{aligned}$$

This number is obtained by observing that $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{S}_A - \{Cert_{X,Y}\}$, $\mathcal{V}_1 \cup \mathcal{V}_3 = \mathcal{S}_A - \{Cert_{X,B}\}$, $\mathcal{V}_2 \cup \mathcal{V}_3 = \mathcal{S}_A - \{Aut_{A,Y}\}$ and $\mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3 = \mathcal{S}_A$ and applying (5) to compute the probabilities that these sets are contained in $View_A$. Alternatively, but in this case less efficiently, $conf(Aut_{A,B})$ could be computed by determining for each of the $2^7 = 128$ subsets of $\mathcal{S}_A$ whether $Aut_{A,B}$ can be derived, and adding these probabilities. Note that in this example we have $conf(Aut_{A,Y}) = 0.89 > p(Aut_{A,Y}) = 0.8$.

Example 4.7. The example of Figure 5 (left) illustrates a new problem, namely certification cycles, which is quite likely to occur in a large-scale practical scenario and is easily handled by our model. Here the four minimal certification paths are $A - X - B$, $A - Y - B$, $A - X - Y - B$ and $A - Y - X - B$. They correspond to the minimal sets $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$ of the previous example plus the additional set

$$\mathcal{V}_4 = \{Aut_{A,Y}, Trust_{A,X,1}, Trust_{A,Y,1}, Cert_{Y,X}, Cert_{X,B}\}.$$

The confidence level of $Aut_{A,B}$ results in $conf(Aut_{A,B}) = 0.8276$. Note that, as could be expected, the additional certificate $Cert_{Y,X}$ improves the confidence value for $Aut_{A,B}$ only marginally.

Example 4.8. For the first time in this section we now consider a recommendation by extending the previous example with the statements $Trust_{A,Y,2}$ and $Rec_{Y,X,1}$ where $p(Trust_{A,Y,2}) = 0.7$ and $p(Rec_{Y,X,1}) = 0.9$ (see Figure 5, right side). Note, however, that the statements $Trust_{A,Y,1}$ and $Trust_{A,Y,2}$ are not independent because $Trust_{A,Y,2}$ implies $Trust_{A,Y,1}$. The confidence value of $Aut_{A,B}$ is $conf(Aut_{A,B}) = 0.838$ which is a noticeable improvement over example 4.7.

4.5 A discussion of the examples

Let us briefly summarise the examples. The confidence values we have chosen may appear to be quite low, but they illustrate our points more clearly than values close to 1.

Except for example 4.5, the set of statements of the examples of this section (examples 4.3, 4.4, 4.6, 4.7, and 4.8) increases monotonically. As a consequence,

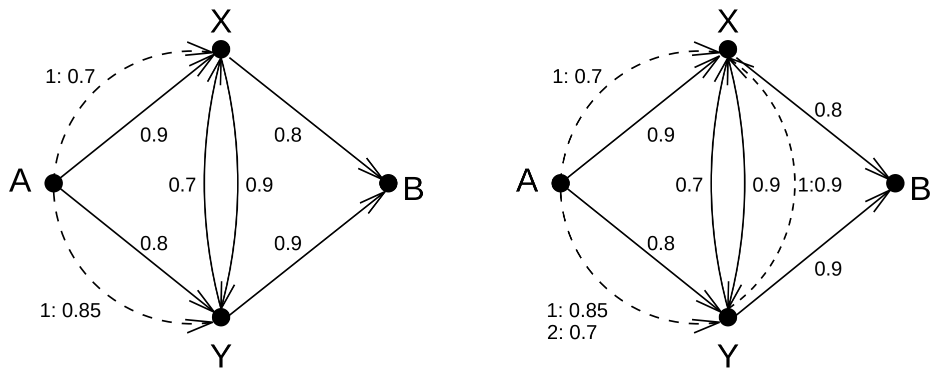

Figure 5.

the confidence value of $Aut_{A,B}$ increases monotonically. This appears to be a desirable property of a model that does not contain "negative" statements such as revocations. The single certification path of example 4.3 gives a confidence value of only 0.504. By adding a parallel path (example 4.4), the value is increased to 0.8076. Adding a further certificate $Cert_{X,Y}$ increases the value to 0.825 whereas the certificate $Cert_{Y,X}$ only adds marginally to the confidence value (0.827). The recommendation $Rec_{Y,X,1}$ increases the confidence value to 0.838.

The most significant increase of confidence in $Aut_{A,B}$ is achieved by introducing an independent certification path. In contrast, all subsequent improvements appear to be rather insignificant because no additional independent entities are involved and the minimal subsets from which $Aut_{A,B}$ can be derived overlap significantly. Our model gives a natural explanation of this fact. The confidence values would increase more significantly if a third independent certification path were introduced. For example, introducing a third path $A-Z-B$ in example 4.4, namely the additional statements $Aut_{A,Z}$, $Trust_{A,Z,1}$ and $Cert_{Z,B}$ (with confidence parameters 0.8 each), would increase $conf(Aut_{A,B})$ to 0.906, and a further such independent path would increase it to more than 0.95. This illustrates the importance of independent certification paths.

4.6 Efficiency considerations

For very large views with many certification paths (or, more precisely, with many minimal subsets of $\mathcal{S}_A$ from which $Aut_{A,B}$ can be derived) this described algorithm is infeasible. The examples illustrate that certain minimal subsets can be discarded without much effect on the confidence value of $Aut_{A,B}$. Hence one can efficiently obtain a good approximation to the confidence value which is guaranteed to be pessimistic, i.e. on the "safe side".

For this purpose, it may be useful to perform a sensitivity analysis to find those minimal subsets with small marginal impact on the confidence value of $Aut_{A,B}$. This is suggested as a problem for future research.

As mentioned in Section 4.3, Alice's probabilistic view must, in any realistic application, be specified by a rather small number of parameters. This means that only a small number of dependencies can be modelled. A reasonable simplification is achieved by specifying the (infinite) list of parameters $p(Trust_{A,X,i})$, for $i = 1, 2, \ldots$, as a function of only one parameter α. For instance, one could define

$$p(Trust_{A,X,i}) = \alpha^i$$

for some $\alpha < 1$.

5 Concluding remarks and open problems

We have proposed a deterministic and a probabilistic model for a user's view of a public-key infrastructure. They include recommendations and confidence parameters for statements. It appears likely to the author that recommendations in the context of a PKI will in future implementations be made explicit. It is perhaps less clear whether confidence parameters will ever be used widely, but in a sufficiently user-friendly and error-tolerant implementation this appears quite possible. One of the applications of using confidence parameters could be for letting certificates expire gradually rather than sharply on a particular date. This could be achieved by letting the confidence parameter decrease with time.

For any model of public-key certification there is an inherent trade-off between the levels of details of a particular scenario that can be captured and the complexity of its specification and analysis. Important open research problems are the design of efficient algorithms and simplifications of the model that result in confidence values with guaranteed accuracy.

An interesting but non-trivial problem, which is the subject of ongoing research, is to incorporate public-key revocation into the model. It must be specified who can revoke a public key. In order to make false revocations unlikely it is perhaps useful to specify for each public key a list of entities authorised to revoke the public key.

A drawback of our model is that users must assign precise confidence parameters to the statements available to them. An interesting extension could be to allow an incomplete specification of the parameters of the model, for instance by specifying intervals rather than exact values for the probabilities, or by only partially specifying the dependencies between the parameters. The goal of such an approach would be to derive upper and lower bounds on the confidence values of statements that are consistent with the partial specification.

Acknowledgements

I would like to thank Jan Camenisch, Germano Caronni, Martin Hirt, Jean-Marc Piveteau and Markus Stadler for interesting discussions, and the anonymous referees for several suggestions for improving the paper. The comments from Joachim Biskup and Dieter Gollmann have been particularly useful. Martin Hirt provided help with the figures and programs for computing confidence values. I am also grateful for the generous hospitality of the Isaac Newton Institute for Mathematical Sciences at the University of Cambridge, where part of this paper was written.

References

1. T. Beth, M. Borcherding and B. Klein, Valuation of trust in open systems, *Computer Security – ESORICS '94*, D. Gollmann (Ed.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1994, vol. 875, pp. 3–18.
2. A. Birell, B. Lampson, R. Needham and M. Schroeder, A global authentication service without global trust, *Proc. IEEE Symposium on Research in Security and Privacy*, 1986, pp. 223–230.
3. C. Boyd, Security architectures using formal methods, *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, 1993, pp. 694-701.
4. M. Burrows, M. Abadi and R. Needham, A logic of authentication, *ACM Transactions on Computer Systems*, vol. 8, no. 1, 1990, pp. 18–36.
5. E. A. Campbell, R. Safavi-Naini and P. A. Pleasants, Partial belief and probabilistic reasoning in the analysis of secure protocols, *Proc. The Computer Security Foundations Workshop V*, IEEE Computer Society Press, 1992, pp. 84–91.
6. S. Chokhani, Towards a national public-key infrastructure, *IEEE Communications Magazine*, vol. 32, no. 9, 1994, pp. 70–74.
7. W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. 22, no. 6, 1976, pp. 644–654.
8. R. Fagin and J. Y. Halpern, Uncertainty, belief, and probability, *Proc. of the Eleventh International Joint Conference on Artificial Intelligence*, August 1989, vol. 2, pp. 1161–1167.
9. W. Feller, *An Introduction to Probability Theory and its Applications*, third ed., vol. 1, New York, NY: Wiley, 1968.
10. J. Glasgow, G. MacEwen and P. Panangaden, A logic for reasoning about security, *ACM Transactions on Computer Systems*, vol. 10, no. 3, 1992, pp. 226–264.
11. V. D. Gligor, S.-W. Luan and J. N. Pato, On inter-realm authentication in large distributed systems, *Proc. IEEE Conference on security and privacy*, 1992, pp. 2–17.
12. T. Hailperin, Probability logic, *Notre Dame Journal of Formal Logic*, vol. 25, no. 3, July 1984, pp. 198–212.
13. B. Lampson, M. Abadi, M. Burrows and E. Wobber, Authentication in distributed systems: theory and practice, Proc. 13th ACM Symp. on Operating Systems Principles, 1991, pp. 165–182.
14. U. M. Maurer and P. E. Schmid, A calculus for secure channel establishment in open networks, *Proc. 1994 European Symposium on Research in Computer Security (ESORICS' 94)*, D. Gollmann (Ed.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1994, vol. 875, pp. 175–192.
15. R. Molva, G. Tsudik, E. Van Herreweghen and S. Zatti, KryptoKnight Authentication and Key Distribution System, *Proc. 1992 European Symposium on Research in Computer Security (ESORICS 92)*, Y. Deswarte, G. Eizenberg, J.-J. Quisquater (Eds.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1992, vol. 648, pp. 155–174.

16. A. Nerode and R. A. Shore, *Logic for Applications*, Springer Verlag, 1993.
17. N. J. Nilsson, Probabilistic logic, *Artificial Intelligence*, vol. 28, no. 1, 1986, pp. 71–86.
18. C.H. Papadimitriou, V. Rangan, M. Sideri, "Designing Secure Communication Protocols from Trust Specifications", *Algorithmica*, 1994, pp. 485-499.
19. P. V. Rangan, An axiomatic theory of trust in secure communication protocols, *Computers & Security*, vol. 11, 1992, pp. 163–172.
20. R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, vol. 21, no. 2, 1978, pp. 120–126.
21. B. Schneier, *Applied Cryptography*, 2nd edition, John Wiley & Sons, Inc., New York, 1996.
22. J. G. Steiner, B.C. Neuman and J.I. Schiller, Kerberos: An authentication service for open network systems, Proceedings of *Winter USENIX 1988*, Dallas, Texas.
23. W. Stallings, *Network and Internetwork Security*, Englewood Cliffs, NJ: Prentice Hall, 1995.
24. P. Syverson and C. Meadows, A logical language for specifying cryptographic protocols requirements, *Proc. IEEE Conf. on Research in Security and Privacy*, 1993, pp. 165–180.
25. J. J. Tardo and K. Alagappan, SPX: Global authentication using public key certificates, *Proc. IEEE Conf. on Research in Security and Privacy*, 1991, pp. 232–244.
26. R. Yahalom, B. Klein and T. Beth, Trust relationships in secure systems – a distributed authentication perspective, *Proc. IEEE Conf. on Research in Security and Privacy*, 1993, pp. 150–164.
27. P. Zimmermann, PGP User's Guide, vol. I and II, Version 2.6, May 22, 1994.
28. ISO/IEC International Standard 9594-8, Information technology – open systems interconnection – the directory, Part 8: Authentication framework, 1990.
29. Privacy enhanced mail (PEM), Internet Request for Comments (RFC) 1421–1424.