Reasoning about Attribute Authenticity in a Web Environment

Thomas Wölfl
University of Regensburg
93053 Regensburg, Germany
Thomas.Woelfl@wiwi.uni-regensburg.de

Abstract

The reliable authentication of user attributes is an important prerequisite for the security of web based applications. Digital certificates are widely used for that purpose. However, practical certification scenarios can be very complex. Each certificate carries a validity period and can be revoked during this period. Furthermore, the verifying user has to trust the issuers of certificates and revocations. This work presents a formal model which covers these aspects and provides a theoretical foundation for the decision about attribute authenticity even in complex scenarios. The model is based on the event calculus, an AI technique from the field of temporal reasoning. It uses Clark's completion to address the frame problem. An example illustrates the application of the model.

Introduction

The authenticity of attributes plays a crucial role for the information security of web based applications. For example, the applicability of asymmetric cryptography which allows encryption or digital signature of electronic messages relies on the authenticity of the used public keys (Menezes, van Oorschot, & Vanstone 1997). It has to be assured that a used key is the one really belonging to the receiver or the signee of the message respectively. The public key is an example of a particular user attribute. Other examples of attributes are biometric reference templates, the possession of a prepaid card or certain access privileges. The authentication of these attributes enables the authorization or the identification of the associated holder. For instance, an authentic privilege authorizes the holder to access a protected resource such as an electronic banking account (Chadwick 2003).

In case of public keys, digital public key certificates are widely used to certify their authenticity. These are exchanged and managed by means of public key infrastructures (PKIs) (Adams & Lloyd 2003). Lopez et al. (2004) regard the usage of digital certificates as the state-of-the-art approach for the authentication of attributes different from public keys, too. The concept of public key certificates is expanded to attribute certificates with the goal to authenticate arbitrary attributes. In this way, a certificate based authentication and authorization infrastructure (AAI) is formed which provides a basis for security services such as timestamping, non-repudiation or single sign on (SSO) in an Internet environment.

However, practical AAI scenarios can be very complex. Each certificate has a certain validity period and, furthermore, certificates can be revoked during this period. Additionally, trust plays a major role since the user authenticating a certain attribute has to trust the issuers of the used certificates. For instance, the verifier of a public key certificate has to trust the issuer to certify authentic public keys. Additionally, it has to be defined which entities are allowed to issue revocations for a given certificate.

This work presents a formal model of certificate based AAIs which is based on the *event calculus* (Kowalski & Sergot 1986), an AI technique from the field of temporal reasoning. In contrast to existing PKI models (cf. related work) it covers the authentication of arbitrary attributes and the important aspects of validity periods and certificate revocations. The model provides a theoretical foundation for a clear decision about attribute authenticity even in complex situations.

Its main part consists of eight axioms formulated by means of first order logic. The AAI is modeled from the perspective of an abstract user Alice. This user adds formulas representing her knowledge about the AAI to the model's axioms and verifies whether attribute authenticity is a logical consequence of the *completion* of this set of formulas. The *consistency* is shown under a certain premise.

The remainder of this paper is organized as follows. The first section explains some AAI concepts and describes the used notation. After this, the core of the model consisting of events, fluents and axioms is presented. An example illustrates the model's application. Finally, we present related work and conclude with a summary.

Certificate Based AAIs

This section describes AAI concepts and sketches the ideas behind the model. Figure 1 shows an example of a digital certificate. These data structures are digitally signed by the certificate issuer and certify the authentic binding of an attribute to the certificate subject. The digital signature protects the authenticity and integrity of the content which allows the exchange of certificates via unsecured open networks. The issuer assigns a validity period for each issued certificate (Adams & Lloyd 2003).

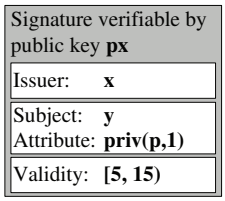

Figure 1: Content of a Digital Certificate

We take the perspective of the user Alice who is interested in the authenticity of another user's attribute. Therefore, Alice retrieves an adequate certificate and checks the following conditions for certificate verification: (1) Does Alice trust the issuer for the certification of the current attribute, i.e. does Alice believe that this issuer certifies authentic attributes? (2) Is the certificate's digital signature valid? To verify this, Alice has to know the authentic public key of the certificate issuer. (3) Is the certificate valid at the current time, i.e. is the certificate unrevoked and is the current time within the validity interval? Alice accepts the certificate if all these conditions are satisfied.

Trust Management (condition 1)

At first, the certified attributes are characterized more detailed. This leads to a novel approach for trust management. The attributes can be divided into two distinct groups, namely properties and privileges. **Properties** have a descriptive character. They consist of a *property type* (e.g. public key) and a *value* (e.g. the value of the key). Examples of property types are the membership in a user group, the possession of a ticket, a biometric reference template or a public key. In the following, a property of type t with value v is represented in this way:

$$prop(t, v)$$

Privileges explicitly describe the rights of an entity. In contrast to properties they can be *delegated* to other entities. They consist of a *privilege type* and the *delegation level* of the privilege. Some examples of privilege types are the access right to bank account x, the right to read and execute file y or the right for admittance to building z. The delegation level is a natural number restricting the maximum length of a delegation chain beginning at the privilege holder: level 1 means that the privilege can be applied but cannot be delegated; level 2 means that the privilege can be applied and delegated via a maximum of 1 instance; level 3 means that the privilege can be applied and delegated via a maximum of 2 instances; etc. This approach is a generalization of Maurer's trust levels (Maurer 1996) for the use with privilege types different from the privilege of issuing public key certificates. A privilege of type t with delegation level i, i.e. the right to delegate the privilege via a maximum of $i - 1$ instances ($i > 0$), is represented in this way:

$$priv(t, i)$$

Based on this distinction, the following approach is used for trust assignment and propagation. Trust in the issuer of a certificate for a property can be regarded as privilege type, namely the *"privilege of certificate issuing for a certain property type t"*. This specific privilege type is denoted by the following function:

$$ci(t)$$

Consequently, digital certificates can be used for the assignment and the delegation of this trust privilege. Alice expresses her own trust in other entities using self-signed certificates, i.e. she issues and digitally signs a certificate using her own private key. Thus, she assigns the trust privilege $priv(ci(t), i)$ with a certain delegation level i to an entity. Depending on the used level, this entity is able to apply that privilege (and issue a certificate for a property of type t) and/or to propagate it by issuing another certificate which contains the same trust privilege (this corresponds to a recommendation). Alice controls the maximum number of delegation steps by setting the delegation level in the self-signed certificate.

A similar approach is used with the certification of privileges. The issuer of a certificate for a certain privilege $priv(t, i)$ has to hold this very privilege itself with a higher delegation level $j > i$. Again, Alice expresses her trust by issuing a self-signed certificate for that privilege type. This certificate sets the beginning of a delegation chain consisting of a number of certificates for the same privilege type. Each certificate issuer assigns the privilege to the next issuer in the chain. Its maximum length (i.e. the number of contained certificates) is restricted by the delegation level in Alice's self-signed certificate and the delegation levels of the subsequent certificates.

This trust management concept allows the assignment of meta-level privileges, concerning the right for certificate issuing (trust), and standard-level privileges used for entity authorization, in a uniform way by means of attribute certificates.

Public Key Authenticity (condition 2)

The verification of a certificate's digital signature requires the authentic public key of the issuer. Of course, digital certificates are used for that purpose. Again, Alice represents the authenticity of initially known public keys using self-signed certificates. These certificates set the beginning of so-called public key certification chains (Adams & Lloyd 2003) where each certificate is used for signature verification of the next certificate in the chain. Using the distinction of attributes described above, public keys are properties. We use the constant κ to denote the property type *"public key"*. Thus, classical trust in the issuer of a public key certificate is expressed by the privilege $priv(ci(\kappa), 1)$.

Certificate Validity and Revocation (condition 3)

It is a straightforward task to verify whether the current time lies within the certificate's validity interval. However, the crucial part is the verification of certificate revocations. First, it has to be assured that revocations for a given certificate are discovered. Second, it has to be defined which entities have the right to revoke that certificate. The first

point describes an availability problem which is out of the scope of the current paper.[1] The second point can also be solved using the certification of trust privileges. An entity has the right to revoke the certificates of issuer x if it holds the privilege $priv(cr(x), 1)$. The function

$$cr(x)$$

denotes the privilege type for the revocation of certificates issued by entity x. This revocation privilege can be assigned by self-signed certificates and propagated by attribute certificates in the described way, too.

Formal Model Based on the Event Calculus

The formal model for reasoning about attribute authenticity is based on a Shanahan's (1999) dialect of Kowalski's and Sergot's (1986) event calculus which is a logic-based formalism for the representation of events and their effects. There are many other temporal reasoning approaches, the most prominent of which is probably the situation calculus (McCarthy & Hayes 1969). We choose the event calculus because it allows the development of a specialized version, whereby a certain number of possible events are predefined and their effects are incorporated into the axioms of the model. Furthermore, it facilitates the realization of a PROLOG program (Nilsson & Małuszyński 2000) for an automated derivation which is a very useful tool, especially in complex situations with numerous certificates and revocations.

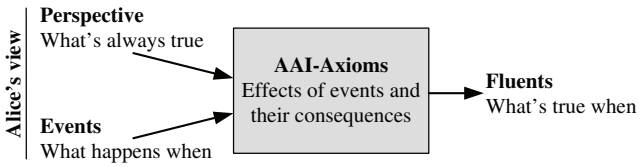

Figure 2: Design of the AAI Model

Figure 2 gives an overview of the model's functionality. Alice describes the observed AAI events and specifies her own perspective (e.g. her own public key). Both parts are represented by atomic formulas and are collected in a set called *Alice's view*. The axioms model the effects of events and their further consequences. They allow conclusions about the validity of *fluents* which are used to represent digital certificates and attribute authenticity.

Fluents, Events and Time Points

Fluents are descriptions that apply to some intervals of time and do not apply to others (Dean, Allen, & Aloimonos 1995). An example is a statement such as "the window is open"; its validity changes with the course of time. In an AAI, there exist two fluents:

$cert(x, px, h, a)$ denotes a digital certificate which certifies that entity h has the authentic attribute a. The certificate is allegedly[2] issued and digitally signed by entity x. The digital signature can be verified by public key px, i.e. the signature is generated by a private key associated to public key px.

$aut(h, a)$ denotes the authenticity of the binding between attribute a and entity h (attribute authenticity).

Next, we describe which events influence the validity of these two fluents. Alice observes three events in an AAI:

$begin(c)$ denotes the event of the beginning of certificate c's validity period.

$end(c)$ denotes the event of the end of certificate c's validity period.

$revokes(r, pr, c)$ denotes the event of revocation of certificate c. The revocation is allegedly issued and digitally signed by entity r. The digital signature can be verified by public key pr.

The $begin$ and end events happen at the two points in time which delimit a certificate's validity interval (recall Figure 1). Since they are contained in the certificate, both are protected by its digital signature. The $revokes$ event is modeled differently, because a revocation can carry a digital signature different from the signature of the revoked certificate, e.g. in case of a delegated revocation. The revocation event happens at the moment when the revocation should take effect.

Finally, we have to decide for a representation of time. The used time model is *dense*, i.e. for each pair of ordered time units there exists a third unit in between, and *bounded*, i.e. there exists a lower bound of time (Gerevini 1997). Thus, time points are represented by real numbers greater than or equal to 0.

Predicates and Axioms

The basic ontology of the event calculus, that is to say the types of things over which quantification is permitted, comprises events, fluents and time points (Shanahan 1999) described in the last section. Going hand in hand with the choice of ontology is the choice of basic *predicate symbols*. The AAI model includes predicates for saying what happens when, for describing the terminating effects of actions, for saying what's always true and for describing what fluents hold at what time. The following list introduces the used predicate symbols and their intended interpretations:

$Happens(e, z)$: Event e occurs at time z.

$Terminates(e, f, z)$: Event e has a terminating effect on fluent f at time z.

$Always(f)$: Fluent f is valid at any time z with $0 \preceq z$.

$HoldsAt(f, z)$: Fluent f is valid at time z.

$Invalid(z_1, f, z_2)$: Fluent f is invalid between times z_1 and z_2, i.e. there exists a point in time lying in the interval $[z_1, z_2]$ where the fluent f is not valid.

[1] We assume the use of certificate revocations which have a negative character saying that a certificate is not valid any more. There exist different theoretical approaches (Micali 1996) (Zhou 2003) using positive revalidation messages periodically saying that the certificate is still valid.

[2] Without verification of the digital signature there exists no evidence that the certificate was really issued by entity x.

$z_1 \preceq z_2$: Time point z_1 is before or equal to time point z_2.

$i \leq j$: Delegation level i is smaller than or equal to level j.

$i < j$: Delegation level i is smaller than level j.

The core of the model consists of a set of axioms relating the predicates together. The set AAI of axioms contains the following eight formulas:

$$HoldsAt(aut(h,a),z) \leftarrow Always(aut(h,a)) \wedge 0 \preceq z \quad \text{(AAI1)}$$

$$HoldsAt(cert(x,px,h,a),z_2) \leftarrow Happens(begin(cert(x,px,h,a)),z_1) \wedge z_1 \preceq z_2 \wedge \neg Invalid(z_1,cert(x,px,h,a),z_2) \quad \text{(AAI2)}$$

$$Invalid(z_1,cert(x,px,h,a),z_2) \leftarrow Happens(e,z_3) \wedge z_1 \preceq z_3 \preceq z_4 \preceq z_2 \wedge Terminates(e,cert(x,px,h,a),z_4) \quad \text{(AAI3)}$$

$$Terminates(end(cert(x,px,h,a)),cert(x,px,h,a),z) \quad \text{(AAI4)}$$

$$Terminates(revokes(r,pr,cert(x,px,h,a)),cert(x,px,h,a),z) \leftarrow HoldsAt(aut(r,prop(\kappa,pr)),z) \wedge HoldsAt(aut(r,priv(cr(x),1)),z) \quad \text{(AAI5)}$$

$$HoldsAt(aut(h,prop(t,v)),z) \leftarrow HoldsAt(cert(x,px,h,prop(t,v)),z) \wedge HoldsAt(aut(x,prop(\kappa,px)),z) \wedge HoldsAt(aut(x,priv(ci(t),1)),z) \quad \text{(AAI6)}$$

$$HoldsAt(aut(h,priv(t,i)),z) \leftarrow HoldsAt(cert(x,px,h,priv(t,j)),z) \wedge HoldsAt(aut(x,prop(\kappa,px)),z) \wedge HoldsAt(aut(x,priv(t,i+1)),z) \wedge 0 < i \leq j \quad \text{(AAI7)}$$

$$HoldsAt(aut(h,priv(t,i)),z) \leftarrow HoldsAt(aut(h,priv(t,j)),z) \wedge 0 < i < j \quad \text{(AAI8)}$$

These formulas (AAI1) - (AAI8) have the form of program clauses (Lloyd 1987).[3] Therefore, the completion described later can be applied to these formulas.

[3]Instead of the clausal notation $A \leftarrow L_1, \ldots, L_n$ the equivalent notation $A \leftarrow L_1 \wedge \ldots \wedge L_n$ is used in order to point out that "," represents conjunction. We use the notational convention that every variable is universally quantified. This means for example for (AAI1) that $\forall a, h, z$ is not noted for the scope of the entire formula.

The axioms represent real world relations of AAI elements. For each axiom, the underlying concepts are explained in the following.

The first axiom (AAI1) says that fluent $aut(h,a)$ is valid at time z with $0 \preceq z$ if it is always valid. Axiom (AAI2) states that certificate fluent $cert(x,px,h,a)$ is valid at time z_2 if it is initiated by the beginning of the associated certificate's validity period at time $z_1 \preceq z_2$ and if it is not invalid between the times z_1 and z_2.

Axiom (AAI3) states that fluent $cert(x,px,h,a)$ is invalid between the times z_1 and z_2 if

1. event e occurs at time z_3 and
2. time point z_3 is before or equal to time point z_4 and both time points lie within the interval $[z_1, z_2]$ and
3. event e has a terminating effect on fluent $cert(x,px,h,a)$ at time z_4.

Axiom (AAI4) states that the end of a certificate's validity period has a terminating effect on the corresponding certificate fluent at any time. Axiom (AAI5) says that a revocation $revokes(r,pr,cert(x,px,h,a))$ has a terminating effect on fluent $cert(x,px,h,a)$ at time z if

1. public key pr is the authentic key of revocation issuer r at time z (it is used for the verification of the revocation's digital signature) and
2. revocation issuer r holds the privilege of type $cr(x)$ with delegation level 1 at time z. $cr(x)$ denotes the privilege type for the revocation of certificates issued by entity x.

Axiom (AAI6) is used for the authentication of properties. The binding of property $prop(t,v)$ to entity h is authentic at time z if

1. certificate $cert(x,px,h,prop(t,v))$ for this property is valid at time z and
2. public key px is the authentic public key of certificate issuer x at time z and
3. certificate issuer x holds a privilege of type $ci(t)$ with delegation level 1 at time z. $ci(t)$ denotes the privilege type for issuing certificates containing a property of type t.[4]

Axiom (AAI7) allows the authentication of privileges. The binding of privilege $priv(t,i)$ with delegation level i to entity h is authentic at time z if

1. certificate $cert(x,px,h,priv(t,j))$ is valid at time z which certifies privilege type t with delegation level j with $0 < i \leq j$ and
2. public key px is the authentic public key of certificate issuer x at time z and
3. certificate issuer x holds a privilege of the same type t with higher delegation level $i+1$ at time z.[5]

Axiom (AAI8) states that privilege $priv(t,j)$ implies privilege $priv(t,i)$ if the delegation level i is smaller than j and greater than 0.

[4]If $t = \kappa$, this corresponds to trust in the issuer of a public key certificate. In this case, rule (AAI6) is equivalent to Maurer's (1996) rule (1) if the time component is neglected.

[5]If $t = \kappa$, rule (AAI7) corresponds to Maurer's (1996) rule (2).

Completion and Attribute Authentication

The model uses Clark's *completion* (1978) to address the well-known *frame problem* (McCarthy & Hayes 1969). In short, the frame problem can be explained as follows: How do we use logic to describe the effects of events without having to explicitly represent all their non-effects (Shanahan 1999)?

The completion can be described as a transformation of program clauses combined with the *unique name assumption* (Baker 1991). It allows Alice to draw negative conclusions based on the lack of positive information. For example, if Alice's view does not contain an event for the beginning of a certificate's validity period, the completion enables Alice to conclude that this event did not happen. The completion, denoted by $Comp$, results from transformation steps as described e.g. by Lloyd (1987).

Now we have established the means for reasoning about attribute authenticity. In a first step, Alice describes her view of the AAI using the $Always$ and $Happens$ predicates. These atomic formulas are collected in the set $View$. Alice is interested in the authenticity of an attribute (a property or a privilege) for a certain entity at a given point in time. The binding of attribute a to entity h is authentic at time z if $Comp(AAI \cup View)$ is *consistent* and if

$$Comp(AAI \cup View) \models HoldsAt(aut(h, a), z)$$

i.e. if $HoldsAt(aut(h, a), z)$ is a logical consequence from the completion of $AAI \cup View$.[6]

Consistency

The consistency of the set $Comp(AAI \cup View)$ is a crucial precondition for the applicability of the model. The following theorem facilitates a reliable decision about the set's consistency.

Theorem 1 (Consistency)**.** *If there exists no revocation for a certificate which is used for the authentication or the authorization of this certificate, then $Comp(AAI \cup View)$ is consistent for any finite set $View$ containing variable-free $Happens$ and $Always$ atoms.*

A proof of that theorem is provided by (Wölfl 2006). The reason for inconsistencies are *revocation cycles*. Consider the following paradox situation: A revocation r for certificate c exists which is used for the authentication or the authorization of the revocation r. It cannot be decided if either c or r is valid: the validity of r implies the invalidity of c which implies the invalidity of r, etc. In case Alice's view does not contain a revocation cycle, the set $Comp(AAI \cup View)$ is consistent according to the theorem above. Again, the PROLOG program (Wölfl 2006) can be used for an automated decision about the existence of a revocation cycle.

[6] A calculus such as the *Complete Systematic Tableau* (CST) (Nerode & Shore 1997) can be used for verification whether $HoldsAt(aut(h, a), z)$ is a logical consequence. However, it turned out that a manual derivation can be long and time-consuming because of the complexity of the modeled domain. Therefore, an automated reasoning system should be used as derivation tool. The work (Wölfl 2006) presents a PROLOG program for that purpose.

Example

The following example illustrates the application of the model. Figure 3 shows five certificates and a revocation. The three certificates on the left hand side are self-signed by Alice (denoted by **a**)[7]. In this way, Alice represents her belief that entity **x** has the authentic property $prop(\kappa, \mathbf{px})$ (a public key) and that **x** has the privilege $priv(ci(\kappa), 1)$ for the issuing of public key certificates. The entity **x** applies that privilege and certifies $prop(\kappa, \mathbf{pb})$ to entity **b** and $prop(\kappa, \mathbf{pr})$ to entity **r**. Furthermore, Alice assigns the privilege $priv(cr(\mathbf{x}), 1)$ for the revocation of **x**'s certificates to entity **r**. That entity revokes the certificate $cert(\mathbf{x}, \mathbf{px}, \mathbf{b}, prop(\kappa, \mathbf{pb}))$ at time 5.

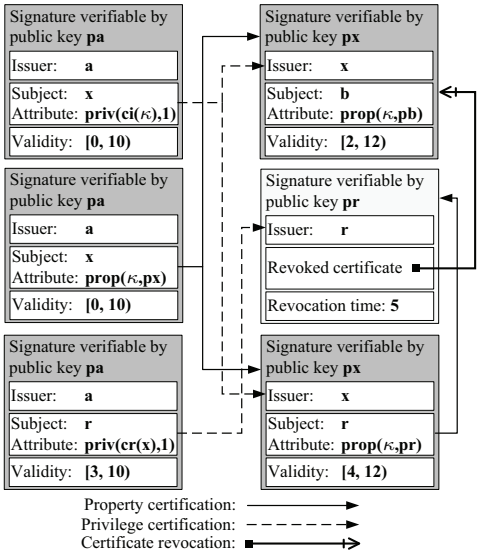

Figure 3: Five Certificates and a Revocation

The shown situation is formally represented by the following set $View$.

$$
\begin{aligned}
View = \{ & \\
& Always(aut(\mathbf{a}, prop(\kappa, \mathbf{pa}))), \\
& Always(aut(\mathbf{a}, priv(ci(\kappa), 2))), \\
& Always(aut(\mathbf{a}, priv(cr(\mathbf{x}), 2))), \\
& Happens(begin(cert(\mathbf{a}, \mathbf{pa}, \mathbf{x}, priv(ci(\kappa), 1))), 0), \\
& Happens(end(cert(\mathbf{a}, \mathbf{pa}, \mathbf{x}, priv(ci(\kappa), 1))), 10), \\
& Happens(begin(cert(\mathbf{a}, \mathbf{pa}, \mathbf{x}, prop(\kappa, \mathbf{px}))), 0), \\
& Happens(end(cert(\mathbf{a}, \mathbf{pa}, \mathbf{x}, prop(\kappa, \mathbf{px}))), 10), \\
& Happens(begin(cert(\mathbf{a}, \mathbf{pa}, \mathbf{r}, priv(cr(\mathbf{x}), 1))), 3), \\
& Happens(end(cert(\mathbf{a}, \mathbf{pa}, \mathbf{r}, priv(cr(\mathbf{x}), 1))), 10), \\
& Happens(begin(cert(\mathbf{x}, \mathbf{px}, \mathbf{b}, prop(\kappa, \mathbf{pb}))), 2), \\
& Happens(end(cert(\mathbf{x}, \mathbf{px}, \mathbf{b}, prop(\kappa, \mathbf{pb}))), 12), \\
& Happens(revokes(\mathbf{r}, \mathbf{pr}, cert(\mathbf{x}, \mathbf{px}, \mathbf{b}, prop(\kappa, \mathbf{pb}))), 5), \\
& Happens(begin(cert(\mathbf{x}, \mathbf{px}, \mathbf{r}, prop(\kappa, \mathbf{pr}))), 4), \\
& Happens(end(cert(\mathbf{x}, \mathbf{px}, \mathbf{r}, prop(\kappa, \mathbf{pr}))), 12) \}
\end{aligned}
$$

[7] Constants are printed using bold fonts.

The $Always$ formulas describe Alice's own attributes. For instance, Alice regards **pa** as her own authentic public key. She is able to make the following conclusions about attribute authenticity. For example, from her perspective public key **pb** is the authentic public key of entity **b** at time 2:

$$Comp(AAI \cup View) \models HoldsAt(aut(\mathbf{b}, prop(\kappa, \mathbf{pb})), 2)$$

The revocation causes that this is not true at time 5:

$$Comp(AAI \cup View) \models \neg HoldsAt(aut(\mathbf{b}, prop(\kappa, \mathbf{pb})), 5)$$

These results can be verified using the mentioned PROLOG program or another (automated) derivation method. The model's potential is developed to the full in more complex situations with numerous certificates and revocations having divers validity periods and revocation times.

Related Work

The main part of related work belongs to the field of formal PKI models. Three significant models are mentioned. Jøsang's (1999) certification algebra which is based on *subjective logic* covers trust and public key authenticity. It allows an agent to compute its opinion about the authenticity of a certified public key.

A seminal paper of Maurer (1996) introduces a deterministic PKI model defining trust levels as natural numbers with clear semantics. The PKI is modeled from the perspective of the user Alice who uses four rules to derive new statements from her collected knowledge. Maurer's work does not cover certificate validity periods, revocations and attributes different from public keys. In contrast to the mentioned approaches the current work covers these aspects.

Marchesini and Smith (2005) propose an extension of Maurer's model. The user confirms the validity of a certificate by means of "validity templates". However, the work does not include a derivation rule for these templates. Therefore, important aspects such as revocation cycles are not analyzed.

Conclusion

This paper presented a model for the decision about attribute authenticity in a web environment which is based on the event calculus. A possible application is the implementation in a logic program which can be encapsulated in software systems and serve as module for the decision about attribute authenticity. This has the advantage that soundness, completeness and termination of the logic program, being crucial aspects for the relying application, can be formally shown.

References

Adams, C., and Lloyd, S. 2003. *Understanding PKI*. Addison-Wesley, second edition.

Baker, A. B. 1991. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence* 49(1-3):5–23.

Chadwick, D. 2003. The X.509 privilege management infrastructure. In Jerman-Blazic, W. S. B., and Klobucar, T., eds., *Proceedings of the NATO Advanced Networking Workshop on Advanced Security Technologies in Networking*. IOS Press.

Clark, K. L. 1978. Negation as failure. In *Logic and Data Bases*, 293–322. Plenum Press.

Dean, T.; Allen, J.; and Aloimonos, Y. 1995. *Artificial Intelligence: Theory and Practice*. Addison-Wesley.

Gerevini, A. 1997. Reasoning about time and actions in artificial intelligence: Major issues. In Stock, O., ed., *Spatial and Temporal Reasoning*. Kluwer Academic Publishers. 43–70.

Jøsang, A. 1999. An algebra for assessing trust in certification chains. In Kochmar, J., ed., *Proceedings of the Network and Distributed Systems Security Symposium*.

Kowalski, R., and Sergot, M. 1986. A logic-based calculus of events. *New Generation Computing* 4(1):67–95.

Lloyd, J. W. 1987. *Foundations of Logic Programming*. Springer, second edition.

Lopez, J.; Oppliger, R.; and Pernul, G. 2004. Authentication and authorization infrastructures (AAIs): A comparative survey. In *Computers & Security*, 578–590. Elsevier.

Marchesini, J., and Smith, S. 2005. Modeling public key infrastructures in the real world. In Chadwick, D., and Zhao, G., eds., *Proceedings of the 2nd European PKI Workshop (EuroPKI2005)*, number 3545 in Lecture Notes in Computer Science, 118–134. Springer.

Maurer, U. 1996. Modelling a public-key infrastructure. In Bertino, E., ed., *Proceedings of 1996 European Symposium on Research in Computer Security (ESORICS96)*, number 1146 in Lecture Notes in Computer Science, 325–350. Springer.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence 4*, 463–502. Edinburgh University Press.

Menezes, A. J.; van Oorschot, P. C.; and Vanstone, S. A. 1997. *Handbook of Applied Cryptography*. CRC Press.

Micali, S. 1996. Efficient certificate revocation. Technical Report MIT/LCS/TM-542b, Massachusetts Institute of Technology.

Nerode, A., and Shore, R. A. 1997. *Logic for applications*. Springer, second edition.

Nilsson, U., and Małuszyński, J. 2000. *Logic, Programming and PROLOG*. John Wiley and Sons, second edition.

Shanahan, M. 1999. The event calculus explained. In Wooldridge, M., and Veloso, M., eds., *Artificial Intelligence Today*, number 1600 in Lecture Notes in Artificial Intelligence, 409–430. Springer.

Wölfl, T. 2006. *Formale Modellierung von Authentifizierungs- und Autorisierungsinfrastrukturen*. Deutscher Universitäts-Verlag. (Formal modelling of authentication and authorization infrastructures).

Zhou, J. 2003. Efficient signature validation based on a new PKI. In *Proceedings of E-Commerce and Web Technologies: 4th International Conference, EC-Web Prague, Czech Repbulic*, number 2738 in Lecture Notes in Computer Science, 94–103. Springer Verlag.