

# How to Incorporate Revocation Status Information into the Trust Metrics for Public-Key Certification

Kemal BICAKCI  
Bruno CRISPO  
Andrew S. TANENBAUM

In order to validate a digital signature, there should be an authentic link between the corresponding public key and its owner. In a traditional PKI, the trust associated with this link is expressed in binary either by 0 or 1. Alternatively, several authors have proposed trust metrics to evaluate the confidence afforded by a public key. However their work has a static point of view and does not take into account the issue of public key revocation. In this paper, we show how to incorporate the revocation status information into the trust metrics for public key certification for the cases of both the single path and multiple paths certifying the same public key. To achieve our goal, we use a tailored form of a vector of trust model recently proposed. This would allow us to reason formally about when there is a need to check revocation status and how reliable the revocation mechanism should be in a given security application.

- digital signature
- revocation
- trust metrics
- public key certificates
- PKI

## 1

### Introduction

It is commonly agreed that the ability to provide non-repudiation services is a fundamental prerequisite for electronic commerce (e-commerce) applications. Digital signatures, first suggested by Diffie and Hellman in 1975 [1] and first implemented by Rivest, Shamir and Adleman in 1978 [2], are equivalent of handwritten signatures in the digital world and provide not only non-repudiation but also authentication and data integrity services.

However the distinctive property of digital signatures is non-repudiation, which implies that even the recipient himself should be unable to generate the signature otherwise the sender can deny sending by claiming that the signature was generated by the recipient himself. That is why digital signatures cannot be implemented securely using secret-key cryptography where the sender and the recipient have exactly the same power since they share the same secret key.

In public-key cryptography, the aforementioned problem is solved mathematically using the difficulty of a number-theoretic problem. The basic idea is to have two keys instead of one. One of these keys is called the *private key* available only to the signer, used to sign a message and the other one is the *public key* available publicly to everyone who wants to verify a digital signature. Broadly speaking, the security of this two-key solution depends on the computational difficulty of generating the private key given the public key. This security problem can be reduced to the difficulty of a mathematical problem (e.g., factoring large integers as in RSA [2], calculating the discrete logarithms as in DSS [3]).

Additionally, for secure operation, there should be a secure binding between the public key and the signer's identity (a binding between private key and public key is given indirectly), otherwise an adversary is able to forge signatures by publishing a public key claimed to be owned by a wrong identity. First suggested by Kohnfelder [4], this binding is generally provided by *public key certificates*, which are signed messages specifying an identity and the corresponding public key.

Public Key Infrastructure (PKI) provides protocols, services and standards in order to employ the public key certificates securely and effectively. Public key certificates and therefore PKI are also required for establishing confidential channels via public key encryption schemes but in this paper our focus would be on PKI in the context of digital signatures and non-repudiation services.

This work is based on an earlier work 'How to incorporate revocation status information into the trust metrics for public-key certification', Proceedings of the 2005 ACM Symposium on Applied Computing (SAC'05), © 2005 ACM.  
[doi.acm.org/10.1145/1066677](https://doi.org/10.1145/1066677)

With the public key certificates, in order to trust the public key of a user, we need to trust the public key of the authority issuing the certificate(s) as well as the security of its certificate issuance policy. In a traditional PKI model, the trust relationship between any two entities is expressed in binary simply by either 0 or 1 i.e. 0 means not trusted and 1 means trusted. In this basic model if somebody trusts a certificate authority which issues a certificate for a public key, then this public key is trusted.

However, several authors [5, 6] claimed that trust can vary in a range and it should be possible to assign trust (confidence) values between 0 and 1 thereby it becomes possible to model for instance the increase in the trust if multiple paths certifying the same public key is utilised. How much trust improvement can be achieved using multiple paths is a question they have tried to answer by proposing different kinds of trust metrics.

On the other hand, none of these approaches consider the revocation problem. In order to establish the authenticity of the public key, we should also pay attention to the issue of public key revocation.

Revocation means invalidating the public key before its expiration date. Revocation is required for instance when there is a suspect that the private key has been stolen. How can the verifier assure that the public key was not revoked at the time of signing (or verifying)? The previous work implicitly assume that there would be no revocation at all or in a fully-trusted way there is always a fresh check verifying that the certificate is unrevoked. But we know that revocation happens and not rarely. Moreover fresh revocation check is expensive and not practical most of the time.

Our primary goal in this paper is to incorporate revocation status information into the trust metrics for public key certification. In fact, this research problem was considered as an interesting but non-trivial open problem in both [5] and [7]. By doing this it becomes possible to answer questions like:

- If we check the revocation status and see that the public key is not revoked, what would happen to the trust metric?
- What if public key is revoked?
- How would the freshness of revocation status information affect the trust metric?
- What do we lose if the revocation is not checked at all?

The problem of revocation is well-studied and various technical solutions have been proposed. However in all of the previous studies, they start from an assumption. They assume that revocation status is always checked. In contrary, we observe that sometimes revocation is not checked in practice. In this paper we also want to formally discuss the reasons of this real-life behaviour and give some directions regarding when somebody really needs to check the revocation status and how recent this check should be.

For clarity, we would like to emphasise that proposing a new trust metric for public key certificates is not the goal of this paper. Instead, we use prior work on trust metrics e.g. [5, 8, 7, 9] and propose a method to incorporate revocation status information into it.

The rest of the paper is organised as follows. [Section 2](#) gives some background information. [Section 3](#) explains our extended PKI model which includes the revocation. [Section 4](#) shows how one can assign the weights and confidence values in our PKI model and when one should check the revocation status. This section also addresses the case of multiple certification paths. [Section 5](#) mentions some open problems and concludes.

## 2

### Background

In this section, we give some preliminaries on PKI models, trust metrics, vector model of trust and certificate revocation.

## PKI Models

Maurer has proposed two models for a public key infrastructure [5]. He called the first one as the deterministic model which is very similar to what we have presented as the traditional model in the introduction section. However this model takes into account not only certificates but also recommendations. Recommendations are again signed statements like certificates but they have a more general meaning. One can specify the trustworthiness of another entity in a particular context and assign different recommendation levels. For instance a recommendation of level 1 means that the recommended entity is trusted to certify others and a recommendation of level 2 means that the entity is trustworthy in recommending others for certification. As Maurer suggested, certificates can be regarded as recommendations of level 0. Although the topic of recommendation has a great importance, we do not discuss it here further and concentrate on the simpler model involving only the certificates.

Let us now see the fundamental rule of deterministic model when we exclude recommendations:

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

What this equation says is basically if  $A$  trusts somebody  $X$  and  $A$  can authenticate\*  $X$  and when there is a certificate issued by  $X$  to  $Y$  then this implies that  $A$  can authenticate  $Y$  ( $A$  trusts the certificate of  $Y$  when the three statements on the left side are satisfied).

Obviously, that this simplified form does not say something new. Nevertheless it is important in the sense that the probabilistic model can be best explained using it. We use the deterministic model to explain also a probabilistic model.

The deterministic model would not be satisfactory to model a PKI from a user’s point of view. As first presented by Zimmermann [6], trust is not something binary and can range from marginal to ultimate\*\*. As a matter of fact, intermediate degrees of trust is all possible.

Maurer’s second model is a probabilistic model where the confidence value of a certificate is calculated as the product of confidence parameter of authentication of entity issuing the certificate (denoted by  $p(Aut_{A,X})$ ), confidence parameter of trust on the entity issuing the certificate (denoted by  $p(Trust_{A,X})$ ), and confidence parameter of the certificate issued (denoted by  $p(Cert_{X,Y})$ \*\*\*). Confidence value and parameters are in the range of 0 to 1 in the following example.

**Example 1.** Let us suppose that

$$p(Aut_{A,X}) = 0.8, p(Trust_{A,X}) = 0.9, p(Cert_{X,Y}) = 0.7.$$

Then we have

$$conf(Aut_{A,X}) = p(Aut_{A,X}) \times p(Trust_{A,X}) \times p(Cert_{X,Y}) = 0.8 \times 0.9 \times 0.7 = 0.504.$$

If this confidence value would not be enough to perform the transaction, it is possible to have multiple (parallel) paths certifying the same public key and increase the confidence value since the confidence level in case of multiple paths adds up. We will briefly mention how the confidence value can be calculated if multiple paths are used in [subsection 4.4](#). For interested readers, we recommend the original paper [5] for more details. See also [10] for an easy-to-understand explanation of why multiple paths are useful and desired.

In the example above, the first two confidence parameters are assigned by the user and the third one is assigned by the entity issuing the certificate. Notice the difference between the confidence value of the certificate and confidence parameter of the certificate assigned by its issuer. The latter can be thought as another way of specifying certificate classes. As discussed in [11], CA companies like Verisign issue certificates in different classes and these classes are assigned depending on the reliability of the method used to identify the certificate owner. If the certificate issuer is fully trusted, the confidence value of a certificate would have the same value as its confidence parameter assigned by the issuer.

\* It is also possible to think that trusting somebody means authenticating him in the first place. How would you trust somebody whom you can not authenticate? But in this model these two things are differentiated.

\*\* However, PGP is flaky because of the way these trust levels are defined and used as discussed in [7] and [9].

\*\*\*In Maurer’s terminology, the term ‘parameter’ is used when the confidence level is assigned and the term ‘value’ is used when the confidence level is derived.

## Trust metrics

The term 'trust metric' can be defined as the measure of amount of trust attached to something. This 'something' can simply be anything: a person, a file, a URL, a country etc. Public key certificates are just one other application for trust metrics.

In the Maurer's probabilistic model of PKI, probabilities as parameters of subjective belief (denoted as confidence levels) are used as the trust metric for public key certificates. Other than this one, there are all sorts of different ways to represent trust values. In the following, we will mention only one other and leave the survey of the rest to [12].

Assigning confidence parameters between 0 and 1 to various trust statements has raised questions among other researchers. They argue that there is an ambiguity in the semantics and there is no easy way to determine the value of these parameters in real world. For instance, Reiter and Stubblebine [7] suggested a different approach to assign such values and to use numeric labels as a trust metric representing the amount of money the certificate issuer is liable if the authentication of the corresponding public key fails.

It might be true that this insurance-type of trust metrics fits better to real world and most of us understands better if money is the concern. However we stick ourselves to use Maurer's metric since the way Maurer's confidence levels are used can include different approaches because

- It is always possible to convert a value in one trust metric to a value in another metric when the metrics have a finite range. For example, money can always be translated in confidence values by normalising the value to range [0-1].
- These confidence parameters can be based on past experience. For instance it can reflect the risk associated or the reputation of the certificate issuer.
- For some applications, the parameters can be automatically assigned by the system based on some published security policy (this policy can be updated dynamically). If this can be done, the end-user does not need to bother assigning them.

On the other hand if we look at the trust problem from the reverse angle we see that for a wide range of security applications available today, it would be very convenient to set a minimum confidence value required (as a part of the aforementioned security policy) so that the decision whether the trust established would be enough to perform the transaction is not left to the end user.

## Vector model of trust

Trust is something difficult to measure, compare and combine. For a better reasoning of trust, Ray and Chakraborty proposed a vector model to formulate trust in a very recent work [13]. In their model, trust is defined as a vector of several parameters each of which contributes to the overall trust but not in same amount.

As this trust model suggested, one important characteristic of trust is the 'propensity of trust' which means that two trusters may assign two different trust values even when all the factors influencing the trust has exactly the same value. The main reason of this phenomenon is that the truster may assign different weights to different factors during the evaluation of the trust value. The authors have introduced the concept of *trust policy vector* to capture this characteristic. In [section 3.2](#), we will see one example of this policy vector which is simply a vector of weight values.

## Certificate revocation

As we said, the status of a public key certificate can change during its lifetime due to unexpected events (i.e. loss of private key, etc.) thus a solution to reflect this possible change should be found.

A straightforward solution is to use instant or short-lived certificates which have a very short lifetime therefore eliminates the requirement of revocation check. Since this method requires

the users to frequently communicate with the CA to get new certificates, it can not be always used.

For long-lived certificates (with a validity period of months even years), a separate mechanism for revocation check needs to be implemented. By checking revocation information, the verifier will know if the certificate(s) he is going to rely upon is still valid or it has been revoked.

The simplest and most widely used mechanism to implement revocation is *certificate revocation lists*: CRLs. With CRL, the revocation authority issues periodically a signed and time-stamped list of the serial numbers of all the certificates that have been revoked at the time of issuing. To reduce the required bandwidth and most important to reduce the need for on-line connectivity, CRLs should not be issued too frequently. However, increasing the interval between two subsequent CRLs increase also the probability that a certificate that has been reported as compromised by its owner has not yet been published and distributed in the list.

*Delta CRLs* and *Certificate Revocation Trees (CRTs)* [14] are two different improvements to basic CRLs. They succeed in decreasing the bandwidth of revocation information.

Since most of the time, the verifier is not interested in knowing which certificates have been revoked but rather checking if the certificate he just received has been revoked or not, a new mechanism, *On-line Certificate Status Protocol (OCSP)* [15] has been introduced.

With this mechanism, the verifier can query a designated server about the status of a specific certificate. The server always online will reply by signing a statement which asserts the fresh status (valid or revoked) of the certificate in question. The obvious disadvantage of this method is the requirement of on-line connectivity.

## 3

### Our PKI model

To incorporate revocation status information into the trust metrics for public key certification, we extend Maurer's PKI model. Like the original model we have two cases:

#### Deterministic model

As expected, incorporating revocation into the deterministic model is the easier one. We simply modify the rule given in equation 1 as follows:

$$\forall X, Y: \text{Aut}_{A, X1}, \text{Trust}_{A, X1}, \text{Cert}_{X1, Y} \vdash \text{CondAut}_{A, Y} \quad (2)$$

$$\forall X, Y: \text{Aut}_{A, X2}, \text{Trust}_{A, X2}, \text{NotRev}_{X2, Y} \vdash \text{UnRev}_{A, Y} \quad (3)$$

$$\forall X, Y: \text{CondAut}_{A, Y}, \text{UnRev}_{A, Y} \vdash \text{Aut}_{A, Y} \quad (4)$$

Again these equations state the common sense. Conditional authentication denoted by '*CondAut*' means that authentication will be completed only if revocation status is checked as equation 3 says and public key is confirmed not to be revoked.

We observe a couple of inherent weaknesses of this deterministic model:

- This model does not distinguish between revoked public key and the public key without any check of revocation.
- This model does not say anything about the freshness of revocation check and reliability of revocation information.
- The authority used to authenticate the public key might be different than the authority for getting the revocation information. In fact, these two authorities should not be the same otherwise there might be some security vulnerabilities [16]. Again this model does not take into account the different trust values for these two different authorities.

## Probabilistic model

The model introduced in this paper takes into account only the confidence value of the certificate, the result of Maurer's formula denoted by  $\text{conf}(A \rightarrow Y)$ . The trust metric and the method used to calculate this value really do not matter. The method can be either Maurer's method or any other one. Our model is flexible in the sense that it allows us to embed other models and other metrics.

Now we would like to introduce our extended probabilistic PKI model which contains the revocation status information. In our model the confidence value for the public key certificate is assigned explicitly without revocation. We would like to take into account also the revocation therefore in a similar manner we have a confidence value for the revocation information. While doing this, the confidence value for revocation is overloaded also by freshness constraints in our model. For instance a confidence value of 1 would mean that the revocation authority is fully trusted and the revocation information is gathered securely exactly at the time when it is needed. No later, not before.

Now we have two confidence values, one for the public key certificate and the other is for the revocation information of the public key. How can we combine these two values in a reasonable manner? This is where vector of trust model introduced in section 2.2 comes into the scene.

**Definition 1.** The trust vector  $(A \rightarrow Y)$  for our model has two dimensions. One of them is  $\text{Conf}_A$  (confidence value of public key certificate) and the other is  $\text{Conf}_R$  (confidence value for revocation information).

$$(A \rightarrow Y) = (\text{Conf}_A, \text{Conf}_R) \quad (5)$$

**Definition 2.** The trust policy vector  $W$  for our model has two dimensions. One of them is  $W_A$  (weight for checking authenticity of certificate) and the other is  $W_R$  (weight for checking revocation).

$$W = (W_A, W_R) \quad (6)$$

Then the normalised trust relationship between A and Y can be expressed as

$$(A \rightarrow Y)^2 = W \odot (A \rightarrow Y) \quad (7)$$

The symbol  $\odot$  denotes vector multiplication.

**Definition 3.** The value of normalised confidence value for the public key certificate is in the range  $[0, 1]$  and is defined as

$$v(A \rightarrow Y)^2 = W_A \times \text{Conf}_A + W_R \times \text{Conf}_R \quad (8)$$

if  $W_A \times \text{Conf}_A + W_R \times \text{Conf}_R > 0$

$$v(A \rightarrow Y)^2 = 0 \quad (9)$$

if  $v(A \rightarrow Y)^2 \leq 0$ .

The normalised trust value given above is not allowed to be below zero in order to alleviate the calculation of the overall confidence value when multiple paths exist. We will discuss this issue in [subsection 4.4](#). Up to this point, our usage of vector of trust model is very similar to what has been proposed by Ray and Chakraborty [[13](#)]. We only tailor it in two aspects:

- The trust lies within  $[0, 1]$  not within  $[-1, 1]$  in our model. We believe this difference is not so important.
- The sum of all elements in the trust policy vector should always be 1 in [[13](#)]. However this is not always the case in our model. This property is captured in the following definition.

**Definition 4.** The trust policy vector has a length of 1 (the sum of  $W_A$  and  $W_R$  is equal to 1) when the public key is unrevoked and 0 (the sum of  $W_A$  and  $W_R$  is equal to 0) when the public key is revoked. Because if there is revocation this negates the trust for public key certificate.

# 4

## Discussion

After introducing our extended PKI model, we would like to discuss how the trust policy vector is assigned and how one can determine the confidence values for revocation. In this section, we also show how our model can help to decide when to check revocation status as well as how one can compute the overall confidence value in case when multiple certification paths are employed.

### Determining the trust policy vector

Since our trust policy vector with a specified length has only two dimensions, it would be sufficient to present a methodology for the value of one of the weights.

N umber of revoked certificates

$$W_R = \frac{\text{Number of revoked certificates}}{\text{Number of all certificates}} \quad (10)$$

N umber of all certificates

In our model as equation 10 says, the weight for checking revocation information ( $W_R$ ) is fixed to be equal to the ratio of revoked certificates to the total

number of certificates issued so far. Here, our rationale is that this ratio gives us a good estimate for the probability that a given certificate is revoked.

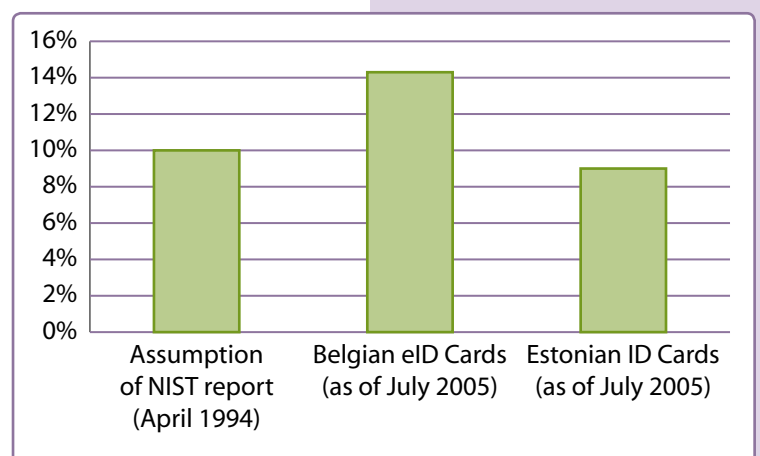
We would like to state that this ratio can change from system to system and from application to application. There are number of parameters that can affect this ratio. For instance, it would be different ratios in applications where public keys support credit card transactions and in applications where public keys are used for IPSec connections. Therefore, while calculating the ratio above, it is crucial to consider only the certificates that can be used in the specific domain.

We have searched the previous work on revocation to see the possible values this ratio can take in different circumstances. Back in 1994, a highly-cited NIST report prepared by MITRE corporation [17] made the assumption of 10% for the certificate revocation rate while they were estimating the total cost of installing the federal portion of the PKI.

In contrast to what most people believed in 1990's and because of reasons which we do not discuss in this paper, wide-scale PKI deployments were not boosted as expected. Only in recently, citizens of some European countries like Estonia and Belgium met with PKI in the form of national identity cards. Together with the classical information all ID cards hold, the chip in the new electronic national ID card contains a digital certificate which allows to generate digital signatures, equivalent to handwritten signatures by law. This would enable a lot of useful applications ranging from online tax declaration to contract signing on the Internet.

As of July 2005; in Estonia 1.64 million ID cards have been issued since January 2002 [18] whereas in Belgium, since April 2003 roughly 1.1 million ID cards have been issued [19]. As illustrated in **Figure 1**, out of total number of activated and unexpired IDs issued so far, 9.06%

**Figure 1.** Certificate revocation rates in different environments



and 14.45% of them were either revoked or suspended in Estonia and Belgium, respectively [18, 19].

The figures above support our argument which states that the information of revocation rate extremely depends on the context. One reason the rate in Belgium is higher than the one in Estonia might be the relative newness of the Belgium eID project.

We believe the information regarding the ratio of revoked certificates is essential and in our PKI model, we require CA or RA (Revocation Authority) to publish with the revocation list (or the other revocation mechanisms) not only the list of revoked certificates but also the ratio of revoked certificates or at least the total number of current valid issued certificates so user can compute the ratio and assign weights accordingly.

**Example 2.** Suppose instant certificates as discussed in subsection 2.4 are used. That means there is no revocation and the weight of checking revocation information is simply zero.

**Example 3.** Assume that the ratio of revoked certificates is 0.1 in a given security application, we assign value of 0.9 to  $W_A$  and 0.1 to  $W_R$  when there is no revocation. As we have stated in Definition 4, if the certificate has been revoked, this has to negate the effect of certification and therefore  $W_R$  should be assigned to (-0.9) if this is the case.

If a more-fine tuning of weights is desirable, this is also possible by specifying the value of  $W_R$  as a function of time passed since the certificate has been issued. If there is not any extraordinary event such as a breakthrough in the cryptanalysis of digital signature algorithms, it is reasonable to expect that the increase in the number of revoked certificates is linear as illustrated in Figure 2. That is why the probability that the public key is revoked and the associated weight are strictly dependent on the time the revocation is checked. This can be best illustrated with an example:

**Example 4.** Suppose a certificate has been issued on January 1st, 2005 and will expire after one year. We need to validate a signature and therefore check the revocation status of a public key on October 1st, 2005 after 9 months the certificate has been issued. The probability that the certificate has been revoked can be found as  $0.10 \times (9/12) = 0.075$  using Figure 2. With this probability, we can assign  $W_R = 0.075$  and  $W_A = 0.925$ . Of course these weights are valid when there would be no revocation.

### Assigning confidence values for revocation information

Next, we would like to discuss another issue in our model, assigning confidence values for revocation information. As we have mentioned in subsection 3.2, the meaning of confidence value for revocation is overloaded in our model because it does not involve only the trust relationship between entities but also the reliability of revocation mechanism in use. Suppose for a moment that the revocation authority is fully trusted but the latest CRL it has issued is not a fresh one, do you put full trust on the revocation information?

It is a rule of thumb in our model that the confidence value if CRL-type methods are in use is lower than the value if OCSP is used. It is also possible to differentiate between fresh CRL and stale CRL by assigning different confidence values even when the authority issuing the lists is the same.

### When to check revocation status

We believe our model explains why revocation is sometimes not checked in real-world. Users believe intuitively that for their security applications it is sufficient to check only the authenticity of the public key certificate since the confidence

Figure 2.

A typical graph showing the change in revocation rate with respect to time.

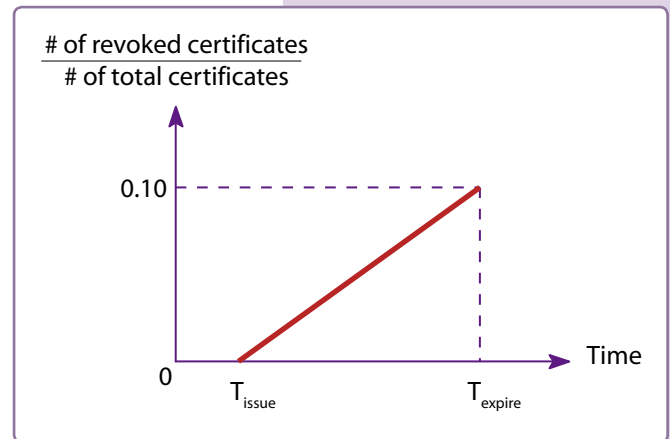
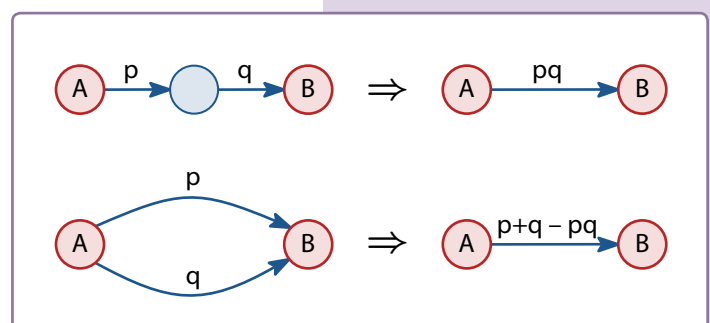


Figure 3.

Combination rules of Maurer's probabilistic PKI model



value of the public key certificate has a much greater weight (the probability of revocation is low). In fact, it is true that some of the applications really do not have stringent trust requirements. But we expect that when certificates start supporting higher-value transactions, using our model people can realise that even when public key certificate is fully trusted, the overall confidence value is not adequate and therefore there is a need to check the revocation status.

We need revocation check definitely when the minimum confidence value required is above  $W_A$ . In case it is lower than  $W_A$ , revocation check should again be performed if the confidence value of certificate is lower than 1 and we would like to reach the minimum confidence value required.

## The case of multiple paths

One of the motivations in proposing different trust metrics for public key certification is to model quantitatively the increase in trust when more than one certification path are used. As we said earlier, our discussion in this paper for incorporating revocation status information into the trust metrics is orthogonal to the prior work on trust metrics. In our extended PKI model, to compute the overall confidence value afforded by multiple paths, one should first use the method proposed in this paper to calculate the normalised confidence values for each individual certification path. Once the values of all certification paths are available, the second step would be to calculate the overall confidence value using one of the methods already available in the literature. To exemplify, [Figure 3](#) illustrates the rules in Maurer's probabilistic PKI model to combine the confidence values of serial and parallel paths. If our extended probabilistic model is used,  $p$  and  $q$  in [Figure 3](#) represent the normalised confidence values incorporating also the revocation information as discussed in [subsection 3.2](#).

We illustrate the general idea with following example:

**Example 5.** Suppose there are two parallel paths certifying the same public key and  $W_R$  is equal to 0.1 when there is no revocation. The confidence values of authentication and revocation are 0.9 and 0.8 for the first and second certification path, respectively. What would be the overall confidence value if the first path is revoked and the second one is not?

For the first path,  $p = W_A \times Conf_A + W_R \times Conf_R = (0.9 \times 0.9) + (-0.9 \times 0.8) = 0.09$ .

For the second,  $q = W_A \times Conf_A + W_R \times Conf_R = (0.9 \times 0.9) + (0.1 \times 0.8) = 0.89$ .

Finally, the overall confidence value is  $p + q - pq = 0.89 + 0.09 - 0.89 \times 0.09 \approx 0.97$ .

# 5

## Conclusion and future work

In this paper, we have considered the problem of certificate revocation in the context of trust metrics for public key certification. We have tackled this problem by extending Maurer's PKI model with a tailored form of a vector of trust model. Our model was not abstract, we were able to justify our model by showing how we can assign values properly to various parameters of the model i.e. the weights and the confidence value for the revocation information.

This extended version of our paper also addresses the interesting issue of multiple paths and the case when one of these paths is revoked. The list below shows challenging future works remaining:

- Trust metrics having a finite range can be embedded into our model by normalising them to the range  $[0, 1]$ . But we need to consider also the trust metrics with an infinite range.
- There is definitely a need to search further for the ratio of revoked certificates and its distribution over time in different security applications.
- More work is required to establish the minimum confidence value required for the public key certificate. This policy work seems to be the most challenging one.

## REFERENCES

1. Diffie W. and Hellman M. (1976), 'New directions in cryptography', *IEEE Transactions on Information Theory*, 1976, vol. 22, pp. 644–654.
2. Rivest R.L., Shamir A. and Adleman L. (1978), 'A method for obtaining digital signatures and public-key cryptosystems', *Communications of the ACM*, 21(2), pp. 120–126.
3. National Institute of Standards and Technology (2000), *Digital Signature Standard (DSS)*, Technical Report FIPS PUB 186-2, Information Technology Laboratory, Gaithersburg, MD 20899-8900, January.
4. Kohnfelder L.M. (1978), *Towards a Practical Public-Key Cryptosystem*, MIT Bachelor's thesis.
5. Maurer U. (1996), 'Modelling a public-key infrastructure', *Proceedings of the ESORICS'96*, LNCS 1146, Springer-Verlag, pp. 325–350.
6. Zimmermann P. (1994), *PGP User's Guide*, vol. I and II, version 2.6.
7. Reiter M. and Stubblebine S. (1999), 'Authentication metric analysis and design', *ACM Transactions on Information and System Security*, 2(2), pp. 138–158.
8. Beth T., Borcherdig M. and Klein B. (1994), 'Valuation of trust in open networks', *Proceedings of the ESORICS'94*, LNCS 875, Springer-Verlag, pp. 3–18.
9. Kohlas R. and Maurer U. (2000), 'Confidence valuation in a public-key infrastructure based on uncertain evidence', *Proceedings of the PKC'00*, LNCS 1751, Springer-Verlag, pp. 93v112.
10. Burmester M. and Desmedt Y. (2004), 'Is hierarchical public-key certification the next target for hackers?', *Communications of the ACM*, 47(8), pp. 68–74.
11. Levi A. and Koc C.K. (2001), 'Risks in email security', *Communications of the ACM*, 44(8), pp. 112.
12. Abdul-Rahman A. (2004), *A Framework for Decentralised Trust Reasoning*, PhD Thesis, University of London; <http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/thesis-final.pdf> (27 May 2005).
13. Ray I. and Chakraborty S. (2004), 'A vector model of trust for developing trustworthy systems', *Proceedings of the ESORICS'04*, LNCS 3193, Springer-Verlag, pp. 260–275.
14. Kocher P. (1999), *A Quick Introduction to Certificate Revocation Trees (CRTs)*, Technical Report, Valicert.

## ACKNOWLEDGMENT

We thank to members of Security Group in Vrije Universiteit for interesting discussions. Our research is supported by SecureE-Justice project funded by EU within the Sixth Framework under the contract IST-2002-507188.

15. Myers M., Ankney R., Malpani A., Galperin S. and Adams C. (1999), *X.509 Internet Public Key Infrastructure – Online Certificate Status Protocol– OSCP*, RFC 2560, IETF.
16. Crispo B. and Lomas M. (1996), 'A certification scheme for electronic commerce', *Proceedings of the 1996 Security Protocols Workshop*, LNCS 1189, Springer-Verlag, pp. 19–32.
17. Berkovits S., Chokhani S., Furlong J.A., Geiter J.A. and Guild J.C. (1994), *Public Key Infrastructure Study: Final Report*, produced by the MITRE Corporation for NIST.
18. *Estonian ID Card Project* (2005); [www.id.ee](http://www.id.ee) (27 July 2005).
19. *Belgium eID Project* (2005); [godot.be](http://godot.be) (27 July 2005).