

MASTER'S THESIS

Dynamic access control

22nd October 2010

Emiel Hollander

Supervisors

dr.ir. Maurice van Keulen (University of Twente)

dr. Virginia Nunes Leal Franqueira (University of Twente)

ir. Anton Boerma (Excellence Group)

ir. Richard Scholten (Excellence Group)



UNIVERSITY OF TWENTE.

Nec me pudet fateri nescire quod nesciam.
*I am not ashamed to confess that I am ignorant
of what I do not know.*

Marcus Tullius Cicero
Tusculanae Disputationes (book I, section 60)

Center on the wide horizon
Focus on the galaxy
Sweep away your expectations
And recognise your enemies

Shirley Manson
Garbage - Afterglow

Voorwoord

Dit is het op-een-na-laatste werk dat ik oplever in het kader van mijn opleiding technische informatica aan de Universiteit Twente. Niet het allerlaatste; dat is immers de presentatie, die ik op het moment van schrijven nog niet gegeven heb.

Tijdens de periode dat ik student was heb ik veel gedaan, veel geleerd en veel mensen leren kennen. Graag wil ik iedereen met wie ik een leuke tijd heb gehad hiervoor bedanken. Ik ga er niet aan beginnen om namen te noemen; het zijn er te veel en ik zou ongetwijfeld mensen vergeten.

Daarnaast wil ik graag nog een aantal mensen in het bijzonder bedanken omdat zij een bijdrage hebben geleverd aan dit afstudeerverslag. Allereerst mijn begeleiders: Maurice en Virginia vanuit de Universiteit Twente, en Anton en Richard vanuit de Excellence Group. De overleggen die we hebben gehad vond ik altijd nuttig en de samenwerking was erg plezierig. Bedankt!

Naast mijn begeleiders heeft nog een aantal mensen mij geholpen. Jo heeft mij van nuttig advies voorzien bij het opstellen van mijn vragenlijst en het verwerken van de evaluatiegegevens. Dankzij deze gesprekken en de boeken die ik van haar mocht lenen kon ik mijn evaluatie naar een hoger niveau brengen. Ze heeft ook het verslag proefgelezen. Brenda, Jochem en Barry hebben, voordat de daadwerkelijke evaluatie begon, het gehele evaluatieproces doorlopen om te kijken of er nog onduidelijkheden of fouten in voorkwamen. Aan de hand van hun opmerkingen heb ik de evaluatie verbeterd. Jochem heeft daarnaast nog een aantal extra mensen geregeld die deelnamen aan de evaluatie. Ook Barry heeft het gehele verslag proefgelezen. Mijn vader heeft een groot aantal deelnemers voor de evaluatie geregeld en heeft mij in contact gebracht met beveiligingsexperts. Bedankt allemaal!

Ten slotte bedank ik graag mijn ouders en mijn broertje voor de steun die ik gedurende mijn gehele studietijd gekregen heb. Als er iets aan de hand was kon ik altijd bij jullie terecht. Dankzij jullie heb ik alles uit mijn studietijd kunnen halen wat erin zat. Heel erg bedankt!

Emiel Hollander
Enschede, oktober 2010

Abstract

An increasing number of services require access control. On the web, access control is usually enforced using a combination of username and password. Users are encouraged to choose secure passwords. These secure passwords are very hard to remember, which causes people to write passwords down, re-use the same password or choose a simple password. Our goal is to design an access control system that is easier to use, while still offering the same amount of security.

The main idea behind this research is that not every service needs the same amount of security. It may not be necessary to ask the secure password for every service; for services that require less security, an access control method that is less secure, but easier to use, may be sufficient.

We have built a system that is capable of dynamically determining the access control method or methods that it has to use to ensure sufficient security. When the user requests a service, the system looks up the amount of security that is needed and adapts the used access control methods to this.

The evaluation of this system shows that people appreciate the fact that the system is able to choose easier access control methods for services that do not require a high security level. According to the participants, the dynamic system is easier and more pleasant to use than an access control method based on caller ID, and easier and more pleasant than DigiD with additional SMS authentication. The participants, however, did not find the dynamic system easier or more pleasant to use than username and password. This system is so common and widely-used that it is hard to beat. We do believe, however, that the dynamic system can become better than username and password when users get more accustomed to it, and when some usability problems have been looked into.

Contents

Voorwoord	v
Abstract	vii
Contents	ix
1 Introduction	1
1.1 Users and security	1
1.2 The digital government	2
1.3 Communication channels	2
1.4 Dynamic authentication	3
1.5 Example	3
1.6 Research questions	4
1.7 Approach	5
1.8 Evaluation	5
1.9 About this research	6
2 State of the art	7
2.1 Authentication	7
2.2 Authorisation	10
2.3 Risk-adaptive access control	11
2.4 Conclusion	11
3 Problem formalisation	13
3.1 Definitions	13
3.2 Behaviour	15
3.3 Open questions	17
4 Quantifying trust and security	19
4.1 Credentials	19
4.2 Trust in authentication methods	21
4.3 Probability of discovery	24
4.4 Discoverability	25
4.5 Security of credentials	30
4.6 Combinations of credentials	30
4.7 Conclusion	34
5 Instance identification	35

Contents

5.1 Duplicate detection	35
5.2 Identifying records	36
5.3 String matching	41
5.4 Conclusion	41
6 Making decisions	43
6.1 Deciding on allowing access	43
6.2 Asking for additional credentials	45
6.3 Assessing the needed security level and identity confidence	47
6.4 Conclusion	48
7 Evaluation	49
7.1 General set-up	49
7.2 Prototype	51
7.3 Questionnaire	52
7.4 Results	54
7.5 Evaluation with security experts	63
7.6 Conclusion	64
8 Discussion	65
8.1 Security level for nonexistent answers	65
8.2 Nonexistent users	66
8.3 Typing errors	66
8.4 High security products	67
8.5 Lack of familiarity with DACS	67
9 Related work	69
9.1 Credential-based access control	69
9.2 Trust-based access control	70
9.3 Human factors in access control	71
10 Conclusions	73
10.1 Future work	75
A Instructions	77
B Questionnaire	79
C Evaluation credentials	89
D Evaluation users	91
E Evaluation results	93
List of definitions	97
List of symbols	99
Bibliography	101

1

Introduction

Access control is annoying. More and more websites request that you register before you can use any of the offered services. When calling a company to request a change in your subscription, they will ask you to identify yourself first.

It may seem like a hassle to perform access control, but it is necessary. We need to identify the user, so that we can attribute the changes or requests to the correct user, and verify his identity, so that nothing can be seen or changed by persons who are not allowed to.

Is there a way to make access control less annoying for users, while still maintaining sufficient security? That is what we have set out to do with this research.

1.1 Users and security

Many users have a negative attitude towards security technologies. They see security mechanisms as an obstacle to performing their daily activities. The main cause mentioned is the persistence of intrusion detectors, virus scanners and other security applications to interrupt their current work [22].

Another cause for users to think too lightly about security is that a number of services uses password protection merely to identify users instead of authenticating them for their own protection. For example, online newspapers and Wikipedia use logins only to track users. This has no benefit for the user at all. They do not care if their password is compromised and therefore choose poor passwords [27].

Many security departments think of the users as enemies, that have to know as little as possible about security mechanisms. According to them, users are “inherently insecure” [1]. It is remarkable that the regulations on security, that security departments themselves impose on users, often cause this insecurity. Users often need to choose a password that has a minimum number of characters, uses numbers and

symbols as well as letters, and they need to change this password periodically to another password that has not been used before. Because of this, users choose poor passwords, because those are easier to remember, or write their passwords down.

In short, users do not want to be bothered by security issues and want to put as little effort into security mechanisms as possible. Is it possible to ask less questions to users, have the system identify and verify users as automatically as possible, and still maintain sufficient security?

1.2 The digital government

Over the past few years, governments around the world have increased their efforts to offer more services online [37]. This has reduced the need for citizens to go and visit their government when they have a request or want to access a certain service.

These services, however, must serve the entire population. This means that users of these government services differ enormously in terms of age, languages known, technical competence and availability of technology. Governments must ensure that no group of users is excluded; everybody needs to be able to use these services [7, 37].

This also means that we cannot assume that the user is in possession of advanced authentication equipment, like fingerprint scanners, or is able to remember all kinds of information that is needed to authenticate him. Ideally we would like to be able to authenticate every user with as little effort from this user as possible.

Since governments are handling sensitive information, we also need to make sure that sufficient security is ensured at all times. User authentication may need to be more thorough for some services.

1.3 Communication channels

Authentication mechanisms are usually tightly coupled to a certain communication channel. When we need authentication via the web, we use a combination of username and password. When someone contacting us via the telephone needs to be authenticated, we could ask him for his postal code, house number and birth date.

This approach has drawbacks. An entire communication channel becomes unusable, when an authentication mechanism fails. It may very well be possible to still authenticate a user via other means. Current authentication systems are unable to handle this situation.

When a separate system dynamically assesses what credentials are still needed to achieve a certain level of confidence in the identification, we can decouple the authentication mechanism from the communication channel.

1.4 Dynamic authentication

We envision a system that can obtain data directly from the communication channel to identify a user, but can also ask additional questions to this user when the data from the communication channel does not provide sufficient confidence in the user's identity. The system will act as a spider in the web between users, communication channels and services, making sure there is enough confidence in the identity of every user for each communication channel, authentication mechanism and service used. A schematic layout of the system we envision is given in figure 1.1.

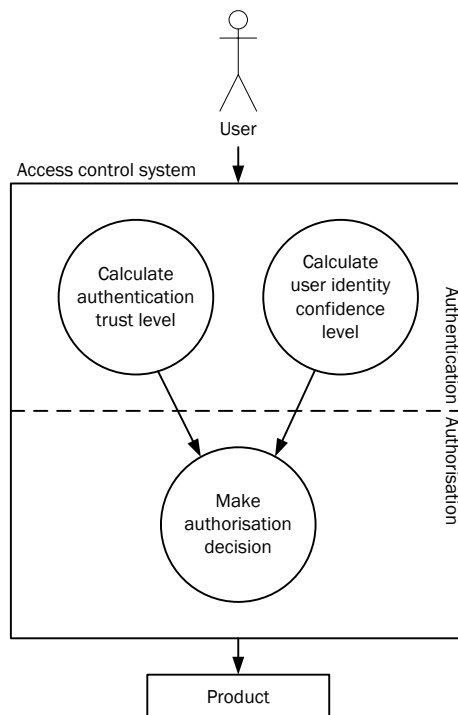


Figure 1.1: Schematic layout of the dynamic access control system

1.5 Example

Alice uses her home telephone to make a telephone call to her municipality because she wants to order a birth certificate. She is connected to the interactive voice response system (IVR) of the municipality.

The IVR system detects that a call is coming in from telephone number 1234. It proceeds to ask Alice which service she wants to access. Via a menu, Alice chooses the service “order birth certificate”.

Next, the IVR system contacts the dynamic access control system (DACs) to request authorisation to access the service Alice chose. It sends all information it has about

the call to the DACS. In this case it only knows the telephone number, so it sends this to the DACS.

The dynamic access control system requests more details about this telephone number from the data store. This system tells the DACS that this telephone number belongs to either Alice, Bob or Charlie. They all live on High Street 1 in Mytown.

Next, the DACS sends a message to the service that Alice wants to access, requesting what information it needs and how many confidence in the verification of the user it wants. The service “order birth certificate” needs a unique reference to a citizen and a verification confidence level of 3.¹

The authentication system checks its list of available identification and verification methods to find a method, or a series of methods, that is capable of delivering a unique reference to a citizen, can deliver this reference with a verification confidence level of 3 and can be used over the telephone.

The telephone number by itself is not enough to obtain a unique reference to a citizen. Policy dictates that verification confidence level 3 can be obtained by asking Alice for her citizen identification number. Further verification using passwords is not necessary for this level. Entering a number is possible using a telephone, so the DACS instructs the IVR system to ask Alice for her citizen identification number. When Alice enters this number correctly, she is granted access to the requested service.

1.6 Research questions

Our main research question is the following.

How can we design a dynamic access control system that takes the information from the communication channel and the requirements of the requested product into account when selecting an appropriate authentication mechanism?

There are several subquestions to ask for this problem.

1. How can we obtain a confidence in the user’s identity that is as high as needed, even when the authentication system is presented with incomplete data?
2. How can we have the authentication system automatically assess the amount of risk involved when allowing a certain user to access a service, and take decisions based on this assessment?
3. Which authentication methods are available, which amount of security can they ensure and which communication channels can they be used with?
4. How can we let the authentication system know what input information and security constraints a product needs, so it can use this when taking decisions?
5. In what ways can the authentication system adapt itself when an authentication method is unavailable?

¹This confidence level is fictitious. How the final system will determine the amount of confidence or risk exactly, and how it will quantify this, is one of the subjects of this research.

1.7 Approach

We first do a literature study. First, we want to find methods to match incomplete identifiers to a record in a data set. Information about this may be found in literature on data integration. This information will be used to obtain a confidence in the user's identity that is as high as possible [24, 30, 38].

Second, we need to find literature that describes how security can be adapted to take dynamic authentication into account. We need to look into research on risk-based access control, security levels and trust management [3, 8, 15, 20, 29, 53]. We also need a survey of different authentication methods to determine the amount of security they can deliver [27, 33, 43].

A list of all governmental services, the input data they need and the amount of risk involved in accessing them is needed to enable the dynamic access control system to make decisions based on the service the user wants to access. We need a way to let the authentication system know about this for each service.

Using all this information we design a dynamic access control system that is able to assess the confidence it can have in a user's identity and is able to make security decisions based on this. This system automatically determines which authentication mechanism is best suited for the combination of communication channel used and service requested. Each service needs to have a method to inform the authentication system of its security requirements, so that the authentication system can make decisions based on this.

When the system determines the best authentication mechanism automatically, it should have no problems finding the best alternative when one of the mechanisms fails. This functionality, however, needs to be taken into account from the beginning.

Finally, we can see whether there are easy options to authenticate persons that have already been authenticated for a product, but want to access a product that requires higher security. The authentication they have done for the product that requires lower security is not enough to directly access the product that requires higher security, but we may be able to re-use the information that this user has already entered. We can use this to show the possibilities of dynamic authentication, that are harder to achieve with current authentication mechanisms.

1.8 Evaluation

To evaluate the research we build a prototype of the proposed dynamic access control system. This system will take input from a communication channel and decide whether this information is sufficient to allow the user access to a certain service, based on requirements from the service and identity confidence obtained from a database.

A group of users will work with this prototype. We would like to find out what they think of this new way of access control, especially compared to access control mechanisms they are already familiar with. After they have finished working with

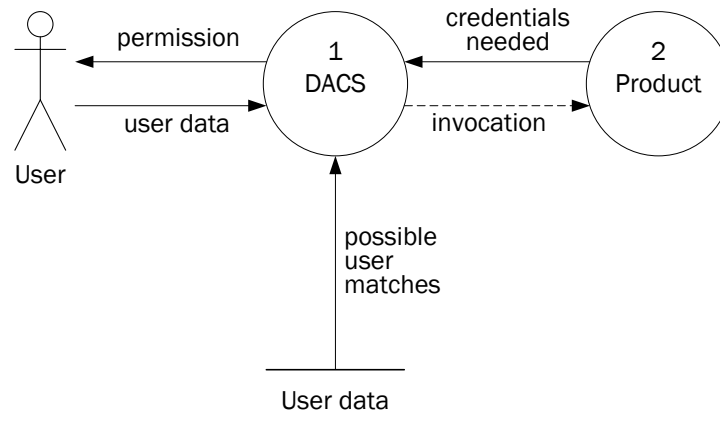


Figure 1.2: Data flow diagram for the dynamic access control system

the prototype, the participants fill in a questionnaire so that we can find out their opinions.

1.9 About this research

The research will be performed for Excellence Group in cooperation with the University of Twente.



The Excellence Group is a company of 50 professionals that provides responsible and innovative ICT solutions for (semi-)public authorities such as municipalities, provinces, health authorities and housing associations. Its goal is to improve service and streamline business processes for their customers. To this end, the Excellence Group has developed an application suite consisting of modular configurable components that enable organisations to implement proven integrated service solutions. For optimal integration, the Excellence Group has developed a variety of adapters for standard software in the domain of public authorities.

UNIVERSITY OF TWENTE. Integrating social and engineering sciences. Developing high tech, with a human touch. It is what the University of Twente is committed to. Through teaching and research at the highest level, and through innovations brought on the market by over 700 spin-off companies. The University of Twente offers degree programmes in fields ranging from behavioural and management sciences to engineering and natural sciences. Research spearheads include nanotechnology, biomedical technology, information technology, governance studies, and learning and cognition.

2

State of the art

Access control emerged in the 1960s to provide isolation of multiple processes on a single system [2, ch. 4]. This kind of access control was enforced by mechanisms built into operating systems, such as preventing processes from writing over or reading each other's memory.

Operating systems also use access control to make sure that users do not access resources that they are not allowed to. An operating system first authenticates a user, and then checks whether this user is allowed to access the resources he is requesting. A typical operating system maintains an access list or a matrix of permissions for this.

Nowadays access control mechanisms have increased in complexity. For example, a user needs access to an application which needs access to middleware which needs access to a database which needs access to a file on a disk. But the underlying principles have not changed. We want to protect resources against the bad guys.

Definition 1 (Access control) *Mechanism that ensures that resources are only granted to those users who are entitled to them [50].*

Before we can control access to resources, we need to know who exactly wants to access them. For this, a number of authentication methods exists. We discuss these in section 2.1. When a user has been authenticated, we can decide whether we grant access or not. Authorisation mechanisms, that ultimately make the decision whether a user is granted access or not, are discussed in sections 2.2 and further.

2.1 Authentication

Definition 2 (Authentication) *The process of confirming the correctness of the claimed identity [50].*

Authentication methods are usually grouped into three groups, as devised by Wood [61]: authentication based on what you know (for example, a password), what you have (for example, a badge) or who you are (for example, a fingerprint).

O’Gorman [43] is not satisfied with this classification. He claims that a password is not strictly *known*, but memorised instead, and can be forgotten. He also says that biometrics, like fingerprints, are not what you are, because a fingerprint or your hair colour, although possibly unique, does not indicate your true self. Every person is a human being; a fingerprint is only a representation of this. Therefore, he proposes the following new labels for these groups.

Knowledge-based authenticators are characterised by secrecy or obscurity, for example a password.

Object-based authenticators are characterised by physical possession, for example a badge or a token.

ID-based authenticators are characterised by their uniqueness to one person, for example a passport or driver’s licence, but also a fingerprint or eye scan.

This classification differs a little from the original classification by Wood. In the original classification, ID-cards and drivers licences belong to the category of object-based authenticators, while O’Gorman puts them in the category of ID-based authenticators.

The classification by O’Gorman will be used in this report. We list a number of authentication methods next. This list is not meant to be complete, but only intends to give examples of the most widely used authentication methods.

2.1.1 Knowledge-based authenticators

Password The password is by far the most widely used authentication method, and has been in use since ancient times; however, it comes with a lot of drawbacks. Schneier describes them aptly: “The problem is that the average user can’t and won’t even try to remember complex enough passwords to prevent dictionary attacks. As bad as passwords are, users will go out of the way to make it worse. If you ask them to choose a password, they’ll choose a lousy one. If you force them to choose a good one, they’ll write it on a Post-it and change it back to the password they changed it from last month.” [52]. These claims are acknowledged by more authors [1, 22, 27].

Secret question The secret question is used for a lot of applications to restore a password after someone has lost it. The question usually concerns a fact that is not well-known, for example, the maiden name of your mother. The large drawback of this authentication method is that, while the fact may not be well-known, it can be obtained. This makes it easy to impersonate someone else [13, 51].

Graphical password Graphical passwords were invented to avoid users choosing insecure passwords and having difficulties remembering them. The idea is that images are easier to remember and that the ability for users to choose secure passwords without a lot of effort is increased. Graphical passwords are either recognition-based or recall-based. A recognition-based graphical password has the user select one or more images, that the user selected earlier to form his graphical

password, from a larger group of images. When using a recall-based graphical password, the user must draw something to authenticate himself, or select a number of spots from an image [54].

2.1.2 Object-based authenticators

One-time password The one-time password was conceived to overcome the problems that exist when using a regular password. They do not have to be remembered, and since they can be used only once, the system will not be compromised when someone learns about a one-time password. One-time passwords can be delivered to the user on a piece of paper, via SMS or e-mail. They can also be generated using a device. This generation is usually a response to a challenge from the server, or it is time-based, when both the server and the device know which password must be delivered at a certain time.

Token A token can be used to authenticate the person possessing it. There are many types of tokens, like cards, badges and small devices. Some of them generate one-time passwords, others contain a chip or magnetic strip that contains identity information. A drawback of tokens is that anyone finding a token can use it to impersonate as the person the token originally belongs to. This can be remedied by protecting the token with a password.

2.1.3 ID-based authenticators

ID-card An ID-card, like a passport or a driver's licence, can be used to verify someone's identity. It can be detected when someone, who is not the owner of the card, uses it.

Biometric features We group all biometric features together since there are so many of them: iris, fingerprints, voice, handwriting and so on. Biometric features are intended to be unique for all people on earth. This makes them very suitable to deliver a unique identification. A drawback of biometric features is that additional equipment is needed in order to do the identification, that may be harder to use for some people [2, ch. 15].

2.1.4 Multi-factor authentication

Multiple authentication methods can be combined to achieve a stronger level of authentication. The combination of authentication methods from different categories is called *multi-factor authentication*. Strictly speaking, asking for both a password and the answer to a secret question is not multi-factor authentication, because both authenticators are knowledge-based. Some examples of multi-factor authentication are listed below.

Bank card When using an ATM to withdraw cash from a bank account, multi-factor authentication is used. This is a combination of an object-based authenticator (the bank card) and a knowledge-based authenticator (the PIN code). This combination

prevents anyone finding a bank card from being able to withdraw funds from the accompanying account.

2.1.5 Implementations

The following authentication methods are implementations of one or more of authenticators mentioned before.

Lightweight Directory Access Protocol (LDAP) Originally, LDAP was designed to offer directory services and functionality to search for persons in an organisation. Because the structure needed for a directory service, that allows users to be allocated to, for example, groups and departments, already exists, it is very easy to exploit this structure to perform authentication and access control [44].

OpenID The aim of OpenID is to provide one authentication service that can be used to log into many websites using only a single password. This eliminates the need to have to remember a password for every site with which you have an account [45].

DigiD The Dutch government has introduced a central authentication method for all its services. The method supports different levels of authentication and provides means to achieve these levels. At the lowest level it requires only a username and password, higher levels require a combination of username, password and a one-time password sent via SMS.

2.2 Authorisation

Definition 3 (Authorisation) *The approval, permission, or empowerment for someone or something to do something [50].*

After a user has been authenticated, authorisation mechanisms limit what this recognised user can do [49]. Authorisation is usually done based on a triple $\langle \text{subject}, \text{operation}, \text{object} \rangle$. The result of the authorisation process is the outcome of a function δ (*decision*) that maps this triple to either authorising or not authorising access.

$$\delta : \text{subject} \times \text{operation} \times \text{object} \rightarrow \{\text{allow}, \text{deny}\} \quad (2.1)$$

We can, for example, define that Bob is allowed to read resource A as follows.

$$\delta(\text{Bob}, \text{read}, \text{Resource A}) = \text{allow}$$

These mappings can be shown in an *access matrix*. This matrix shows what combinations are defined to be allowed.

User	Resource A	Resource B
Alice		write
Bob	read, write	read

Access matrices in operating systems are tightly coupled to the mechanisms used in that operating system. In Unix, for example, the access matrix is stored as protection bits with the protected resource [60].

Since these matrices can become enormous and very cumbersome to manage, actual access control implementations use other methods to store authorisation information. Two examples are access control lists, which store an access matrix by column, and capability lists, which store an access matrix by row [49, 60].

2.2.1 Security policies

Contemporary access control systems are able to assess authorisation decisions using a set of rules, defined using a structured language [57], instead of only an access matrix. These rules may include lookups in an access matrix, but this is not necessary.

We will discuss security policies in more detail in chapter 6. [4, 10, 32, 57, 60].

2.3 Risk-adaptive access control

Risk-adaptive access control methods, also called risk-based access control methods, work in a different way. A risk-adaptive access control method defines a function that can take a number of aspects into account when deciding whether to allow or disallow an action. For this, the method assesses the amount of risk involved or the amount of trust he has in the subject wanting to access an object. Risk-adaptive access control can therefore also be called trust-based access control.

There are currently two major approaches to determine trust: policy-based and reputation-based [11, 12]. Policy-based approaches use certificates, logic and mechanisms with well-defined semantics to make decisions regarding authorisation. Reputation-based approaches, on the other hand, base their decisions on experience they have with the subject, and experiences other systems in their network have had with the subject.

The system we envision is an example of policy-based risk-adaptive access control.

2.4 Conclusion

This chapter explains the underlying principles that are currently used for authentication and authorisation. We have presented examples of existing access control mechanisms. Combinations of these mechanisms can be used for our dynamic access control system. In the next chapters, we will see how we can incorporate these into our system.

3

Problem formalisation

The systems that we have discussed in chapter 2 have static access control mechanisms. They rely on predefined authentication mechanisms. The only exception is risk-adaptive access control, which was treated in section 2.3. We envision a risk-adaptive access control system that is able to dynamically choose the most appropriate authentication mechanism(s) based on the information it has.

This chapter contains a formalisation of the functionality we have in mind for a dynamic access control system. In the following chapters, specific details of this system will be filled in.

3.1 Definitions

To offer dynamic authentication, the system we envision has information about users, communication channels and products. A product is anything that can be offered to a market that might satisfy a want or need [35]. In our case, these products require an amount of security before they can be used.

We define \mathbf{U} to be the set of all users that are known to the system, \mathbf{X} to be the set of all communication channels that are known to the system and \mathbf{P} to be the set of all products that are known to the system.

\mathbf{A} is the set of attributes that a user can have. These can be attributes like name and address, but also, for example, username, password, identification code held in a token, or a biometric signal.

3.1.1 Complete user representations

$R(u) = \{a \leftarrow d_a \mid a \in A, d_a \in D_a\}$ is the representation of a user in the system. This representation contains all attributes and their values for a user. Here, d_a is a value

taken from the domain D_a for attribute a .

$\pi_a(R(u))$ is the projection of attribute $a \in \mathbf{A}$ of the complete representation of user $u \in \mathbf{U}$ in the system. In other words, this is the value d_a for attribute a of user u .

Example 3.1 Suppose $\mathbf{U} = \{\text{Alice}, \text{Bob}\}$ and $\mathbf{A} = \{\text{name}, \text{address}, \text{city}, \text{e-mail}\}$. Note that the entities in \mathbf{U} and \mathbf{A} have no quotes, since these are not strings but references to actual users and attribute entities. An example representation is the following.

$$\begin{aligned} R(\text{Alice}) = \{ & \text{name} \leftarrow \text{Alice}, \\ & \text{address} \leftarrow \text{'High Street 1'}, \\ & \text{city} \leftarrow \text{'Mytown'}, \\ & \text{e-mail} \leftarrow \text{'alice@example.com'} \} \end{aligned}$$

For this representation, $\pi_{\text{city}}(R(\text{Alice})) = \text{'Mytown'}$.

In other words, \mathbf{A} contains all attributes that are available and $R(u)$ contains the values for these attributes for a single user.

3.1.2 Known user representations

$\mathbf{A}^K(u) \subseteq \mathbf{A}$ is the set of attributes that are known for a user u .

$R^K(u) = \{a \leftarrow d_a \mid a \in \mathbf{A}^K, d_a \in D_a\}$ is the known representation of user u in the system.

K in \mathbf{A}^K and R^K is not a variable, but simply refers to “known”.

Example 3.2 Suppose Alice sends an e-mail that is processed by the dynamic authentication system. The system does not know that Alice is mailing so it assigns a temporary name to the user. The system extracts Alice’s e-mail address from this e-mail.

$$\begin{aligned} \mathbf{A}^K(\text{User 1}) &= \{\text{e-mail}\} \\ R^K(\text{User 1}) &= \{\text{e-mail} \leftarrow \text{'alice@example.com'}\} \end{aligned}$$

3.1.3 Product requirements

$\mathbf{A}^N(p) \subseteq \mathbf{A}, p \in \mathbf{P}$, is the set of attributes that are needed for a product p .

$l^N(p) \in \mathbb{R}$ is the level of security needed for a product p .

Similar to the known user presentations, here N in \mathbf{A}^N and l^N is not a variable, but refers to “needed”.

Example 3.3 The product “order birth certificate” needs the name, address and date of birth and security level 3.

$$\begin{aligned} \mathbf{A}^N(\text{order birth certificate}) &= \{\text{name}, \text{address}, \text{date of birth}\} \\ l^N(\text{order birth certificate}) &= 3 \end{aligned}$$

3.1.4 Security level

$l : \mathbf{X} \times \mathbf{A}^* \rightarrow \mathbb{R}$ is the current level of security. This function takes a communication channel χ from \mathbf{X} and a set of attributes from \mathbf{A}^* and maps these to the amount of security that can be deduced from this combination. It is used to calculate a security level based on attributes of a user that are currently known.

Example 3.4 *The e-mail that Alice sent in example 3.2 has been processed further by the dynamic authentication system. The system has also extracted an IP address. It uses this information to calculate the current security level.*

$$\begin{aligned} l(\chi, \mathbf{A}^K(\text{User 1})) &= l(\text{e-mail}, \{\text{e-mail address}, \text{IP address}\}) \\ &= 1.5 \end{aligned}$$

How to calculate this security level is the subject of chapter 4

3.2 Behaviour

Step 1 User $u \in \mathbf{U}$ contacts the system using communication channel $\chi \in \mathbf{X}$ to access product $p \in \mathbf{P}$. Known representation $R^K(u)$ of user u is sent.

Step 2 The system asks product p for needed attributes $\mathbf{A}^N(p)$ and needed security level $l^N(p)$. How to determine the needed security level is discussed in chapter 4.

Step 3 The system calculates the current security level $l(\chi, \mathbf{A}^K(u))$. How this works is discussed in chapter 4. When $l \geq l^N(p)$, go to step 5. Otherwise, go the step 4.

Step 4 The system decides which attributes still need to be known to make $l(\chi, \mathbf{A}^K(u)) \geq l^N(p)$ and that can be asked using channel χ . If no additional information can be asked, authentication fails. Otherwise, proceed with step 3.

Step 5 The system decides whether we have enough information: $\mathbf{A}^N(p) \subseteq \mathbf{A}^K(u)$. When we have enough information, authentication succeeds. Otherwise, go to step 6.

Step 6 The system asks the database whether it has a complete representation $R(u)$ that resembles $R^K(u)$ closely enough. The database delivers a set of representations and confidences, $\check{R} = [0, 1] \times R$, using $f_{\check{R}} : R^K(u) \times R \rightarrow [0, 1] \times R$. The confidence value indicates the amount of confidence the system has that, for a certain $\check{R}(u')$, $u' = u$. In other words, the system finds a list of users that match best with the information that is currently known. How the system does this is discussed in chapter 5.

Step 7 The system deducts $\check{R}(u')$ and $\check{\mathbf{A}}(u')$ for the representation that has the highest confidence. By definition, $R^K(u) \subseteq \check{R}(u')$ and $\mathbf{A}^K(u) \subseteq \check{\mathbf{A}}(u')$. If the confidence is high enough, and the difference in confidence is above a certain threshold for the top results, the system can use $\check{R}(u')$ and $\check{\mathbf{A}}(u')$ during the remaining part of the process instead of $R^K(u)$ and $\mathbf{A}^K(u)$ because there is enough confidence in the identity of the user. If this is the case, authentication succeeds. If the confidence is not high enough the system will simply discard this information and ask extra

questions to the user. If this is the case, continue with step 8. Chapter 6 explains how the system will make this decision.

Step 8 The system determines which question(s) it can ask, using communication channel χ , to obtain $A^N(p) \subseteq A^K(u)$. When multiple questions are possible, the system chooses the one that has the most unique values. How the system decides this is elaborated on in chapter 6. Continue with step 5.

In this context, we can see l as the amount of confidence the system has in the authentication of the user and \check{R} as the amount of confidence the system has in the identification of the user.

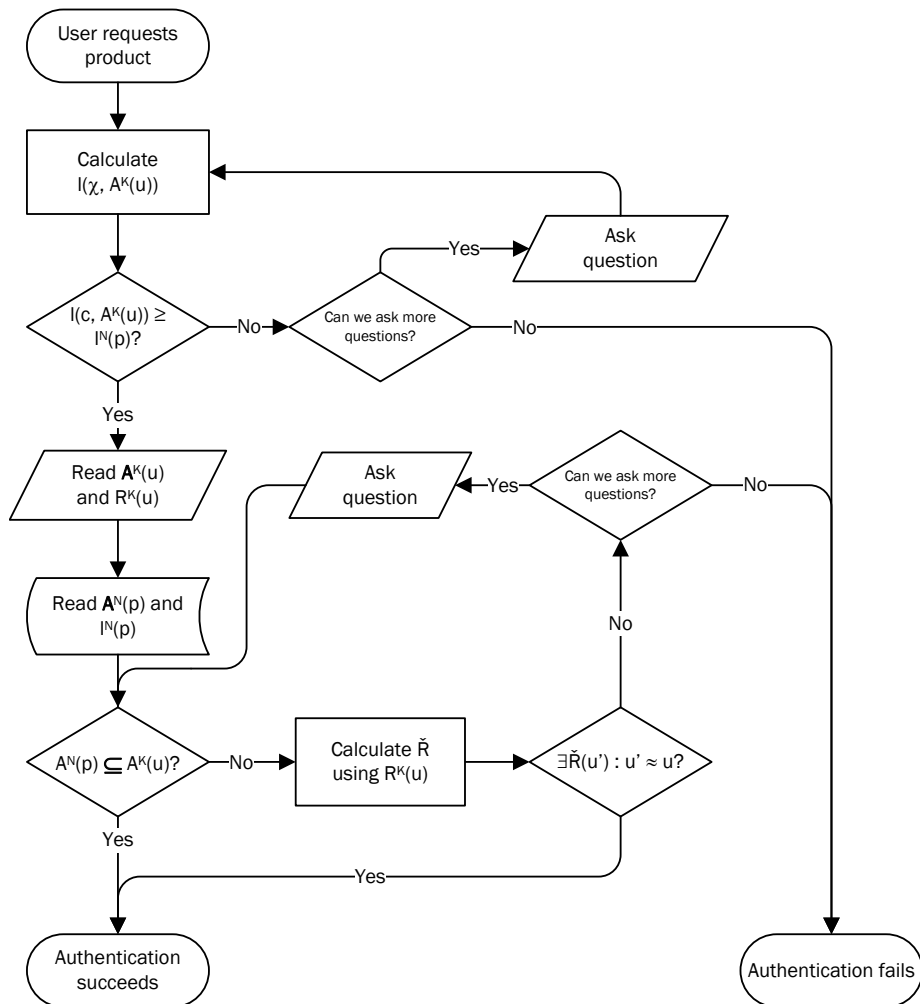


Figure 3.1: Flowchart for the dynamic authentication system

3.3 Open questions

In the design specified above, we have mentioned a number of things that will be elaborated on in the following chapters. We give an overview of these open questions below.

- How do we calculate the current security level $l(\chi, \mathbf{A}^K(u))$? (chapter 4)
- How do we obtain the needed security level $l^N(p)$ for all products? (chapter 4)
- How do we calculate \check{R} based on a known $R^K(u)$? (chapter 5)
- How do we determine when a $\check{R}(u)$ is close enough to $R^K(u)$? (chapter 6)
- How do we decide what additional information is needed to make $\mathbf{A}^N(u) \subseteq \mathbf{A}^K(p)$ and $l(\chi, \mathbf{A}_K(u)) \geq l^N(p)$ with as few questions as possible? (chapter 6)

4

Quantifying trust and security

The notion of quantifying security may sound strange. Usually security is thought of as a binary: either something is secure or it is not [31]. Since we are working with varying authentication methods and varying levels of certainty about the identity of a person, we need dynamic security as well. For some products, a higher level of security may be needed before someone is allowed access, than for others. How exactly can we quantify security? How much security is sufficient for access to a product?

We recognise that the amount of security is affected by the robustness of the authentication mechanism. A password that can easily be guessed offers a lesser level of security than a token that generates one-time passwords. The system needs to conform to the intuitive notion a user may have about the level of security that is provided by the system and needed for a certain product. But how can we assess the robustness of an authentication mechanism?

The contents of this chapter are the basis of the part of our system that calculates the amount of trust we have in the authentication method(s) used, as shown in figure 4.1. We will first look into methods that can be used to quantify the amount of security that we have. We will then apply one method to quantify the amount of trust that a number of authentication methods offer. Finally, we see how to calculate the amount of security for combinations of authentication methods.

4.1 Credentials

Gaining access to products by having the user give credentials that the server assesses is mentioned first in Bina et al. [9]. Examples of credentials are username, password, telephone number and biometric data. Each additional credential contributes to the amount of authentication trust.

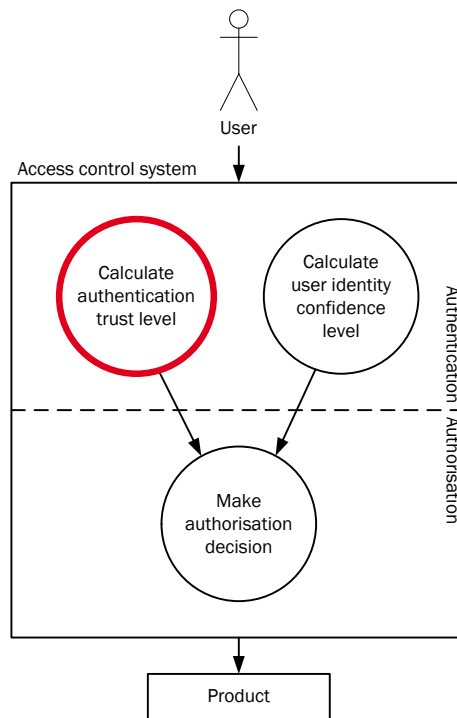


Figure 4.1: Schematic layout of the dynamic access control system

Existing literature usually evaluates authentication methods; for example, the combination of username and password or PIN code. We, however, assess the authentication trust level based on the *credentials* that are known, not on complete authentication methods.

Definition 4 (Credential) *A piece of information that contributes to the successful authentication of a user.*

Using this definition, the authentication method of username and password consists of two credentials: the username and the password. Each credential is also an attribute in the representation of the user, so we can reuse the set of attributes A that we have defined earlier for this.

A credential also often refers to a signed, trusted certificate that is used to verify a user's identity [10, 59]. In our definition, such a certificate can also be a credential, but our definition is less specific. We also need to reason on usernames and other pieces of information. Winsborough et al. [58] define a credential to contain one or more attribute name-value pairs and the public key of the owner. For our research, we define a credential to only contain one name-value pair and no public key. This makes it easier for the system to reason on entered credentials and the user can directly type in the credentials for which this is possible.

4.2 Trust in authentication methods

Methods to quantify trust or security are usually a continuous numerical range or a discrete semantic classification [8, 34]. An example of such a discrete classification is high, medium, low. A lot of governments have already defined several security levels and the minimum security level needed for doing certain transactions [23, 36, 39, 42]. These governments all use discrete classifications.

The New Zealand government identifies different transaction types based on the amount of confidence in one's identity: anonymous, pseudonymous, identified and verified transactions [39]. The government of the United Kingdom on the other hand defines security levels based on the level of damage that can be inflicted when something goes wrong: minimal, minor, significant and substantial damage. These levels of damage are defined using a set of criteria, like the amount of risk to one's personal safety or the amount of financial loss. They also specify the acceptable ways to verify the user's identity for each security level [42]. The government of the United States uses a similar approach, defining assurance levels and several impact profiles [23]. The Dutch government only specifies what kind of authentication is needed to obtain basic, middle or high security levels [36].

Thomas et al. further elaborate on the robustness of authentication mechanisms by proposing *quantified trust levels* [55]. In contrast to trust levels defined by the governments, which are discrete, their trust level is continuous. They have coined the following definition for a quantified trust level.

Definition 5 (Probability of crack) *The probability that an authentication method can be cracked by using random input.*

Let C_{a_1} be the event that the authentication method a_1 is cracked by an attacker. P is the corresponding probability distribution. We define an authentication trust level as:

$$l_{a_1} = -\log(P(C_{a_1})) \quad (4.1)$$

The logarithmic scale is used to create a more human-readable way to represent small probability values. Suppose authentication method a_2 is defective and lets everyone in. The probability that the authentication method is cracked is therefore 1.

$$P(C_{a_2}) = 1 \Rightarrow l_{a_2} = 0$$

Authentication method a_3 has a 0.5% probability of being cracked. Its authentication trust level is:

$$P(C_{a_3}) = 0.005 \Rightarrow l_{a_3} \approx 2.3$$

Example 4.1 *Suppose we use a PIN code as authentication method a_4 . The PIN code has a length of four digits from 0 to 9, and a maximum number of three failed attempts before access is blocked. What is its authentication trust level?*

For a PIN code of length n , the probability that someone can guess the PIN on his first attempt is $\frac{1}{10^n}$. The probability that someone does not guess the PIN on his first attempt, but does guess it on his second attempt, is $(1 - \frac{1}{10^n}) \cdot (\frac{1}{10^n - 1})$.

We can generalise this to the probability that a password-based authentication mechanism a_x , with alphabet Σ , of length n , is cracked after k attempts.

$$P(C_{a_x}) = 1 - \prod_{i=0}^{k-1} \left(1 - \frac{1}{|\Sigma|^n - i} \right) \quad (4.2)$$

The probability that our PIN code of length 4, with a maximum number of three failed attempts, is cracked, is therefore

$$\begin{aligned} P(C_{a_4}) &= 1 - \prod_{i=0}^2 \left(1 - \frac{1}{10^4 - i} \right) \\ &= 0.0003 \end{aligned}$$

The authentication trust level for this PIN is

$$\begin{aligned} l_{a_4} &= -\log(0.0003) \\ &\approx 3.5 \end{aligned}$$

Figure 4.2 shows that the authentication trust level decreases as the probability of crack increases. In other words, authentication methods that are easier to crack receive a lower authentication trust level.

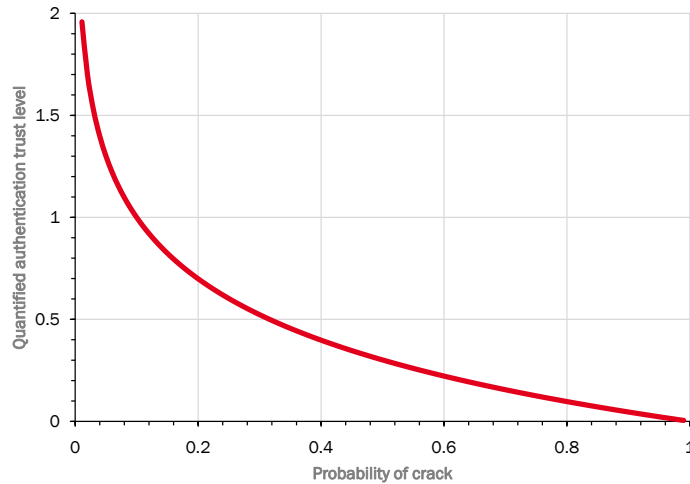


Figure 4.2: Quantified authentication trust level

The drawback of this approach is that it assumes that both the password and the guesses from the attacker are chosen at random. In general, this is not the case. As we have seen before, most people do not choose a password that is completely random. When someone attacking an authentication mechanism is able to make educated guesses instead of random guesses, the probability that he is able to crack the mechanism is much higher.

Also, the probability of guessing a key that has been stored in a token is very low. The amount of trust for such a token, calculated using this method, therefore is very

high; however, someone finding a lost token will have no problem at all in gaining access. This needs to be taken into account when calculating the authentication trust level. This method unfortunately does not do so.

Cheng et al. [15] quantify risk based on the following formula.

$$\text{quantified risk} = \text{probability of damage} \cdot \text{value of damage}$$

They explicitly do not define what damage is because they feel it is the task of the security analyst to determine what exactly the damage is for the organisation and how to value it.

Unfortunately, they also do not give further methods to determine the probability of the damage done. Instead, they claim that “due to the unpredictability of the future, the probability and the value can at best be good enough estimates to compute reasonable quantified risk estimates.”

Sahinoglu [48] bases risk on the combination of vulnerabilities, threats, lack of countermeasures and criticality. He gives the following definitions.

vulnerability A weakness in any information system, system security procedure, internal controls, or implementation that an attacker could exploit.

threat A potential event that will have an unwelcome consequence if it becomes an attack asset.

countermeasure An action, device, procedure, technique, or other measure that reduces risk to an information system.

criticality Indicates the significance of the risk. Criticality is low if risk is of little or no significance, such as the malfunctioning of an office printer, but in the case of a nuclear power plant, criticality is close to 100 percent, because its security is vital for humans.

In Sahinoglu’s model, a vulnerability v_i has an associated probability. The probabilities of all vulnerabilities add up to 1. Each vulnerability has one or more threats t_{ij} . The probabilities of all threats for a single vulnerability add up to 1. Each threat may or may not have a countermeasure cm_{ij} , also with an associated probability. The probability of a lack of countermeasures is $P(\neg cm_{ij})$. The criticality is indicated by c .

Combining this, the formula to calculate the risk is the following.

$$R = \left(\sum_{i,j} P(v_i) \cdot P(t_{ij}) \cdot P(\neg cm_{ij}) \right) \cdot c \quad (4.3)$$

Sahinoglu does not describe how to obtain the probability values for each vulnerability, threat and countermeasure. He only suggests an “educated guess”, using the average of a lower and upper limit of this probability. How to obtain these limits is not described either. Sahinoglu assumes that a security analyst knows how to calculate these values.

He does describe what can be done if purely quantitative data is not available. Sahinoglu suggests to use qualitative attributes and apply a probability to them. We can use, for example, low = 0.25, medium = 0.5 and high = 0.75 to assess the amount of risk involved and use these values in equation 4.3.

4.3 Probability of discovery

During the remaining part of this research, we use the algorithm devised by Thomas et al. to assess the security of credentials. This algorithm allows for an objective assessment of the security that a credential offers by calculating the probability that a credential can be cracked by random guesses.

A drawback of this algorithm, that reduces its accuracy, is that most guesses are not random. Furthermore, a token may contain a long key that is very hard to guess, but, once lost, this token may be used by anyone to gain entry. To overcome this shortcoming, we incorporate a *probability of discovery*, $P(D_a) \in [0, 1]$, in the algorithm.

Definition 6 (Probability of discovery) *The probability that an authentication method can be cracked by informed guesses from the attacker because the probability space of chosen credentials is not uniform or because users write passwords down.*

The probability that a credential is *compromised* is a combination of the probability that it is cracked and the probability of discovery. We call the probability that a credential a_1 is compromised $P(A_1)$, and the probability of discovery for this credential is $P(D_{a_1})$.

Definition 7 (Probability of compromise) *The combination of the probability of crack and the probability of discovery.*

It is clear that the probability that a credential is compromised is always at least the probability that it is cracked. Educated guesses will only make compromising the mechanism easier, so the probability of discovery will increase the probability that a credential is compromised.

When the probability of discovery equals zero, it is clear that $P(A_i) = P(C_{a_i})$. Similarly, when it is certain that a way to gain access to the system can be discovered, so, when $P(D_{a_i}) = 1$, $P(A_i) = 1$ as well. We therefore have to define $P(D_{a_i})$ in such a way that $P(C_{a_i}) \leq P(A_i) \leq 1$ holds when $P(D_{a_i})$ influences the final result.

The probability distribution does not have to be linear. These may differ per access control system. To account for this, we introduce an additional parameter $\alpha \in \mathbb{R}^+$ that can be used to tweak the influence of the probability of discovery on the final probability of compromise. We determine a value for α that is suitable for our access control system when we build and test a prototype.

The equation to calculate the probability of compromise is as follows.

$$P(A_i) = P(C_{a_i}) + (P(D_{a_i}) \cdot P(\bar{C}_{a_i}))^\alpha \quad (4.4)$$

Figure 4.3 shows the influence that the probability of discovery has on the resulting probability of compromise for a number of values of α . In this figure, the probability of crack equals 0.4.

The authentication trust level is also calculated using this new function, so $l_{a_i} = -\log(P(A_i))$ instead of $l_{a_i} = -\log(P(C_{a_i}))$.

Example 4.2 *A system uses the PIN code we assessed in example 4.1. Users are allowed to choose their own PIN code. What is the quantified authentication trust level?*

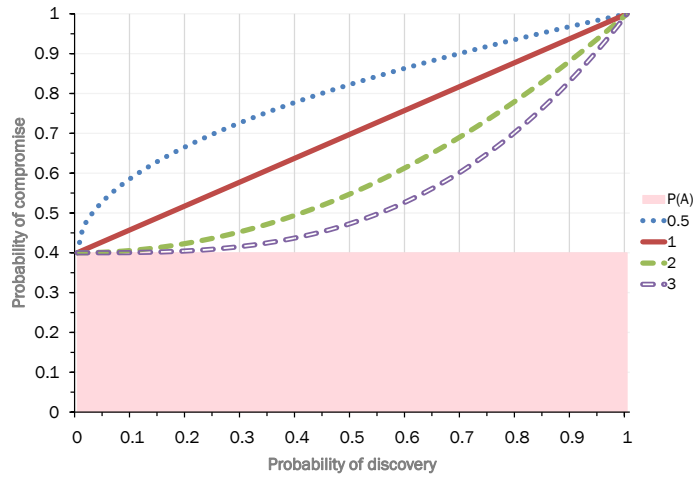


Figure 4.3: Influence of the probability of discovery for different values of α .

For this example we have chosen $\alpha = 1$. From example 4.1 we know that the $P(C_{a_4}) = 0.0003$. When someone chooses his birthday as his PIN code, the probability of discovery is very high. Not everyone, however, chooses such a fact that is easily guessable. We therefore assess the probability of discovery for this credential at high.

How to make a sound assessment of the probability of discovery is treated in section 4.4. For this example, however, this assessment suffices.

$$\begin{aligned}
 P(A_4) &= P(C_{a_4}) + P(D_{a_4}) \cdot P(\bar{C}_{a_4}) \\
 &= 0.0003 + 0.75 \cdot 0.9997 \\
 &= 0.750075
 \end{aligned}$$

Based on this result we can now calculate the authentication trust level of this credential.

$$\begin{aligned}
 l_{a_4} &= -\log(P(A_4)) \\
 &= 0.12
 \end{aligned}$$

For a system where each PIN code is assigned at random we would assess the probability of discovery very low. The trust level for such a system would be $l \approx 1.0$. We assess more credentials in the next section to get a better feeling for what this number exactly means.

4.4 Discoverability

How do we determine the probability of discovery for a certain credential? It is not possible to exactly calculate the probability of discovery. This will always be an estimate. We have defined a classification for the probability of discovery in

Table 4.1: Values for probability of discovery

name	$P(D_a)$
very low	0.1
low	0.25
medium	0.5
high	0.75
very high	0.9

table 4.1. The following sections estimate the probability of discovery for a number of credentials. We will use the classification in table 4.1 for all estimates.

4.4.1 Identification details

Since our system may allow or disallow access also when only identification details, like name, telephone number or e-mail address, are known, we need to specify their probability of discovery as well. This information is usually publicly available, therefore we give these a very high probability of discovery.

4.4.2 Semi-public identification details

Not all identification details can be obtained as easily as someone's name or telephone number. Information like citizen ID or municipality of birth are usually only known to a handful of people in the owner's environment and are not easy to retrieve from public systems. It cannot be justified to give these pieces of information a very high probability of discovery (0.9); therefore, we give these a high probability of discovery (0.75).

4.4.3 Secret question

Schechter et al. [51] have done empirical research on the guessability of secret questions. They have assessed the secret questions of webmail providers AOL, Google, Microsoft and Yahoo! For these sites, they have measured whether secret questions are vulnerable to statistical guessing. They define an answer as statistically guessable when "it is among the five most popular answers provided by *other* participants" of their research. They also assess whether answers can be guessed by the user's partner.

Summing up the results for all tested sites, Schechter et al. found that 13% of all answers are statistically guessable and 22% of all answers can be guessed by the user's partner. The choice of questions has a significant influence on this result. For example, the question "What is your favourite sports team?" is statistically guessable for 57% of the answers.

Considering these results, we assign to the secret question a medium probability of discovery.

4.4.4 Password

The probability of discovery for a password largely depends on how a user chooses his password. When he uses his name, the password is obviously easier to guess than when he uses a random set of letters and numbers.

Yan et al. [62] have researched how easy passwords can be guessed based on the way that the user constructs them. In their experiment, three groups of users were told to construct their password in different ways. The first group of users was told to construct a password that is at least 7 characters long and contains one non-letter. They got no further aid in constructing the password.

The second group constructed their password by using a passphrase. This is a sentence, for example, "It's 12 noon and I am hungry". This sentence is then used to construct the password "I's12&Iah".

The third group was told to randomly construct a password by closing their eyes and randomly picking eight characters from a sheet of paper with the letters A–Z and the numbers 1–9 printed repeatedly on it.

Yan et al. tried four attacks on these passwords. They first performed a dictionary attack. Then they used the words from the dictionaries and permuted them with 0, 1, 2 and 3 digit(s), and made substitutions that are commonly used, like 1 for I and 5 for S. Using this, they again tried to attack the passwords. The third attack exploited known user information, like name, to crack passwords. Finally they tried a brute force attack.

When a user is given no further instructions, besides the minimum length the password needs to have and the fact that it needs to include one non-letter, 32% of passwords can be discovered using the first three attacks. Yan et al. have found that a password that has been constructed using a passphrase is as secure as a randomly chosen string of characters. Of the first, 6% was discovered using the first three attacks, of the latter, 8%.

The probability that a password can be cracked using a brute force attack is estimated by the algorithm of Thomas et al. that we have discussed in section 4.2. We therefore do not discuss this here.

In a similar research, Dell'amico et al. [17] have found that 30% of all passwords of an Italian instant messaging service can be obtained by using a set of dictionaries.

Organisations may choose to give a user a random password and offer no options for him to change it. This way, we can be certain that a user's password is random. Based on the observations above, we give this kind of password a very low probability of discovery. When a user is allowed to change his password, we cannot be certain that it is random anymore. We give this kind of password a medium probability of discovery.

4.4.5 PIN code

Bentley and Mallows [6] investigate the randomness or guessability of a single PIN code within a set of codes. They observe that a PIN code is never chosen

completely random. The probability distribution for PIN codes is therefore not uniform. The probability that such a code is guessed differs for each PIN code and for each distribution. When an attacker has information on the probability distribution, he is able to do educated guesses.

What they do not take into account is that an attacker may be able to do an educated guess because he has information on his victim. When the victim has chosen his birthday as his PIN code, and the attacker knows this birthday, this could be one of the educated guesses the attacker tries first.

Since the results of this research do not suggest otherwise, we choose the probability of discovery for the PIN analogous to these probabilities for passwords. We assess the probability of discovery for a random PIN code at very low and medium for a PIN code chosen by the user.

4.4.6 Graphical password

Davis et al. [16] have researched how easy it is to do educated guesses on graphical passwords. They have tested a graphical password scheme that has been modeled after Passfaces™¹ and a scheme they made up themselves, that they call “Story”. Since only Passfaces™ is used in practice, we disregard the authors’ own “Story” scheme.

Using Passfaces™ the user makes a selection from a number of faces. This selection is his graphical password. When logging in, his selection is shown amongst a number of other faces. When the user selects the correct faces, he is authenticated.

Davis et al. found that the choice of faces is heavily biased based on the race of the user and the attractiveness of the faces used. People tend to select attractive faces from their own race. For certain populations of users, 10% of the passwords that are used by males could be guessed in two tries. Based on this information, we give Passfaces™ a low probability of discovery.

4.4.7 One-time password

A one-time password is usually either acquired from a token that the user possesses (for instance a cell phone), sent by e-mail or retrieved from a previously acquired piece of paper. We assume that the user takes great caution to protect the token or the paper. Therefore we give the one-time password a very low probability of discovery.

4.4.8 Token

The key that is typically used in a token is long enough to not be discoverable at all. Nevertheless, someone may lose his token, making it possible for anyone finding it to access whatever the access control method using the token is protecting. Therefore,

¹<http://www.passfaces.com/>

Table 4.2: Overview of probabilities of discovery

Credential	$P(D_a)$
Biometrics	(4.4.9) very low
Random password	(4.4.4) very low
Random PIN	(4.4.5) very low
One-time password	(4.4.7) very low
Passfaces™	(4.4.6) low
Token	(4.4.8) low
Secret question	(4.4.3) medium
Regular password	(4.4.4) medium
Regular PIN	(4.4.5) medium
Semi-public identification details	(4.4.2) high
Identification details	(4.4.1) very high

a user should take great care in protecting his token. Based on this observation, we give the token a low probability of discovery.

4.4.9 Biometrics

This method using a probability of discovery does not seem to make sense for biometric authentication methods. It makes no sense to assess the probability that a fingerprint or an iris has been discovered. We assume that fingers or eyes are never lost or stolen.

Ballard et al. [5] have done research into the probability that keys, generated from biometric data, are guessed based on auxiliary information, for example, information on how the key is generated, or information on population demographics. In some cases, the probability to guess such a key in a single attempt is 15%. This auxiliary information, however, is usually not available. While 15% may seem high, the chances that this probability is reached are slim.

They also assess the probability that these keys are guessed by using information leaked by templates used to generate the keys. They say that it is important to keep these templates private. Otherwise, the probability of guessing a key in a single attempt could become as high as 22%.

Based on this research, we give biometrics a very low probability of discovery.

4.4.10 Overview

Table 4.2 gives an overview of the probability of discovery for all credentials we have mentioned above. The section in which an authentication contributor is treated is mentioned between parentheses.

4.5 Security of credentials

Table 4.3 gives an overview of quantified authentication trust levels for a selection of credentials. These have been calculated using the method we have described before.

Table 4.3: Quantified authentication trust level for credentials

Credential	$P(C_a)$	$P(D_a)$	l_a
Username (6 letters, chosen by user)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{26^{6-i}}\right)$	0.9	0.05
Last name (8 letters)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{26^{8-i}}\right)$	0.9	0.05
Telephone number (10 digits)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{10^{10-i}}\right)$	0.75	0.12
Citizen ID			
PIN (4 digits, chosen by user)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{10^{4-i}}\right)$	0.5	0.30
Password (8 characters, chosen by user)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{94^{8-i}}\right)$	0.5	0.30
PIN (4 digits, random)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{10^{4-i}}\right)$	0.1	1.00
Password (8 characters, random)	$1 - \prod_{i=0}^2 \left(1 - \frac{1}{94^{8-i}}\right)$	0.1	1.00
One-time password			
Token			
Secret question			
Passfaces™			
Fingerprint			
Iris scan			
Voice recognition			

4.6 Combinations of credentials

Once a single credential cannot deliver a high enough trust level, the system may decide to try additional credentials. It then needs to calculate a new trust level based on the combination of the credentials that have been used.

This new trust level is based on the probability that two credentials, a_1 and a_2 , both fail: $P(A_1 \cap A_2)$. But, since events A_1 and A_2 may not be independent, we cannot simply say $P(A_1 \cap A_2) = P(A_1) \cdot P(A_2)$. When an intruder knows how to crack a password, he may also know how to crack a PIN code. When these two credentials are used together, the probability of both being cracked is not independent.

For dependent events, $P(A_1 \cap A_2) = P(A_1) \cdot P(A_2 | A_1) = P(A_2) \cdot P(A_1 | A_2)$ by definition. The problem is that we cannot calculate $P(A_2 | A_1)$ as this conditional

probability is not known. It is not possible to calculate the influence that the probability that authentication method a_2 has been compromised has on the probability that a_1 will also be compromised.

Thomas et al. [55] have thought of this as well. Their idea is to estimate $P(A_1 \cap A_2)$. To do this, they first define the lower and upper bounds that this function can have. Because we are working with probabilities, the lower bound represents the highest trust while the upper bound represents the lowest trust.

They assume that $P(A_2 | A_1) \geq P(A_2)$ since the probability that a credential is compromised can only *increase* when another credential has been compromised, and will never decrease. Using this assumption we can deduce that $P(A_1) \cdot P(A_2) \leq P(A_1) \cdot P(A_2 | A_1) = P(A_1 \cap A_2)$. This is the lower bound.

Thomas et al. further argue that any combination of credentials is at least as strong as its strongest credential, since an attacker has to go through all credentials to gain access. So, when comparing two credentials, the amount of trust the strongest one offers is $\min(P(A_1), P(A_2))$. This is the upper bound.

We now have a first approximation of the probability that two credentials are compromised.

$$P(A_1) \cdot P(A_2) \leq P(A_1 \cap A_2) \leq \min(P(A_1), P(A_2)) \quad (4.5)$$

4.6.1 Similarity coefficient

To estimate whether the actual value of $P(A_1 \cap A_2)$ is closer to the upper or the lower bound, Thomas et al. describe a *similarity coefficient* $h \in [0, 1]$ that describes the similarity of different credentials. This similarity coefficient is based on the idea that, when two credentials are similar, it is easier to crack both than when they differ a lot. When using two similar credentials, the authentication trust level should not rise as much as when using two credentials that are very different. The similarity coefficient is meant to accomplish this.

Since $P(A_1 \cap A_2) = P(A_1) \cdot P(A_2 | A_1)$ and $P(A_1)$ is known, we can deduce that this similarity coefficient only influences the value of $P(A_2 | A_1)$. This value is not known, but we do know the lower and upper bounds. These can be based on equation 4.5. Since $P(A_2 | A_1) = \frac{P(A_1 \cap A_2)}{P(A_1)}$, the following holds.

$$P(A_2) \leq P(A_2 | A_1) \leq \min \left(1, \frac{P(A_2)}{P(A_1)} \right) \quad (4.6)$$

We can also intuitively see that this equation is correct. When the similarity coefficient equals zero, the probability that credential a_1 has been compromised has no influence at all on the probability that credential a_2 is compromised as well, so $P(A_2 | A_1) = P(A_2)$. On the other hand, when the similarity coefficient equals one, the compromise of credential a_1 has a maximum effect on the probability that credential a_2 is compromised; so, when credential a_1 is compromised, credential a_2 is compromised as well, unless a_2 's probability of compromise is lower than that of a_1 .

Table 4.4: Similarity coefficients for multiple authentication methods (from [55])

Description	Similarity coefficient
Multi-factor authentication	0.1
Authentication methods belong to the same category	0.6
Authentication methods are the same with different parameters	0.95

Based on this, Thomas et al. define the following functions, given the upper and lower bounds of $P(A_2 | A_1)$.

$$\begin{aligned}
p_{A_1} &\equiv P(A_1) \\
p_{A_2} &\equiv P(A_2) \\
\inf(P(A_2 | A_1)) &= f_{\inf}(p_{A_1}, p_{A_2}) = p_{A_2} \\
\sup(P(A_2 | A_1)) &= f_{\sup}(p_{A_1}, p_{A_2}) = \min\left(1, \frac{p_{A_2}}{p_{A_1}}\right)
\end{aligned}$$

These functions, combined with the similarity coefficient h , that describes how close the actual values are to either the lower or upper bound, can be used to define an estimate of $P(A_2 | A_1)$:

$$\begin{aligned}
P_h(A_2 | A_1) &= f_{\inf}(p_{A_1}, p_{A_2}) + h \cdot \Delta(f_{\sup}(p_{A_1}, p_{A_2}), f_{\inf}(p_{A_1}, p_{A_2})) \\
&= p_{A_2} + h \cdot \left(\min\left(1, \frac{p_{A_2}}{p_{A_1}}\right) - p_{A_2} \right)
\end{aligned} \tag{4.7}$$

Using this definition, we can now define $P(A_1 \cap A_2)$ using $P_h(A_2 | A_1)$:

$$\begin{aligned}
P(A_1 \cap A_2) &= P_h(A_2 | A_1) \cdot p_{A_1} \\
&= \left(p_{A_2} + h \cdot \left(\min\left(1, \frac{p_{A_2}}{p_{A_1}}\right) - p_{A_2} \right) \right) \cdot p_{A_1} \\
&= p_{A_1} \cdot p_{A_2} + h \cdot (\min(p_{A_1}, p_{A_2}) - p_{A_1} \cdot p_{A_2})
\end{aligned} \tag{4.8}$$

The combined trust level can then be defined as $l_{\text{comb}} = -\log(P(A_1 \cap A_2))$.

How can we determine h , the similarity coefficient? Thomas et al. suggest a simple approach. They assume that multi-factor authentication is stronger than authentication using only one factor. Based on this assumption, they have decided on a number of coefficients, which are shown in table 4.4. Other methods of determining the similarity coefficient are possible. One option they give is a distance function to calculate the difference between two authentication methods and then map this distance to the similarity coefficient.

4.6.2 More than two credentials

The method we have described above unfortunately only works for the combination of two authentication methods. Extending this method to support three or more authentication methods is not trivial.

We have n authentication methods, a_1, a_2, \dots, a_n . We use the following notation to denote the probability that all authentication methods have been compromised.

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P\left(\bigcap_{i=1}^n A_i\right) \quad (4.9)$$

The probability that they are all compromised is the following.

$$P\left(\bigcap_{i=1}^n A_i\right) = P(A_1) \cdot P(A_2 | A_1) \cdot \dots \cdot P(A_n | A_1 \cap A_2 \cap \dots \cap A_{n-1}) \quad (4.10)$$

This equation contains a lot of probabilities that we cannot calculate. We can, however, extend the approximation, that we originally defined for two authentication methods in equation 4.5, for more authentication methods.

$$P(A_1) \cdot P(A_2) \cdot \dots \cdot P(A_n) \leq P\left(\bigcap_{i=1}^n A_i\right) \leq \min(P(A_1), P(A_2), \dots, P(A_n)) \quad (4.11)$$

On the right hand side of equation 4.10, only $P(A_1)$ is known and the rest of the terms are not. Therefore, we cannot easily adapt the approximation to only a part of this equation, like we have done in equation 4.6. Because of this, we simplify the estimate. We define the following functions.

$$\inf\left(P\left(\bigcap_{i=1}^n A_i\right)\right) = P(A_1) \cdot P(A_2) \cdot \dots \cdot P(A_n) \quad (4.12)$$

$$\sup\left(P\left(\bigcap_{i=1}^n A_i\right)\right) = \min(P(A_1), P(A_2), \dots, P(A_n)) \quad (4.13)$$

Using these equations, we can now define the simplified version of P_h as follows.

$$P_h\left(\bigcap_{i=1}^n A_i\right) = \inf\left(P\left(\bigcap_{i=1}^n A_i\right)\right) + h \cdot \Delta\left(\sup\left(P\left(\bigcap_{i=1}^n A_i\right)\right), \inf\left(P\left(\bigcap_{i=1}^n A_i\right)\right)\right) \quad (4.14)$$

The similarity coefficient now gets a slightly different definition. It is not applied to a pair of authentication methods anymore, but to a set. We cannot simply say that the lowest similarity coefficient can be applied to the entire set. Then, two authentication methods that are very alike also have this lowest similarity coefficient associated with them. They appear to be different and their probability of being cracked is given more weight. This is not what we want. Using the same line of reason we can exclude the highest similarity coefficient.

The average value of all pairwise similarity coefficients is a better estimate of the combined similarity coefficient. We use this as similarity coefficient when we assess the security of multiple credentials.

Example 4.3 *We have three authentication methods: a password, a PIN code and an iris scan. What is the average similarity coefficient when all these methods are used?*

The password and PIN code are the same authentication method with other parameters, so these two have a similarity coefficient of 0.95. All other combinations have a similarity coefficient of 0.1 since they all belong to different categories. In this case, the average similarity coefficient $\bar{h} = 0.24$.

4.7 Conclusion

We have defined a way to quantify the authentication trust level for a credential. Using this method we are able to assess the amount of security that a credential can ascertain.

We have also defined how to quantify this authentication trust level when multiple credentials are used.

The dynamic access control system can use these definitions to quantify the amount trust it has in the credentials used. It can also calculate the new amount of trust it would have when another contributor were to be added, and can use this information choose the best credential to ask for next.

Chapter 4 showed how to assess the amount of security offered by the currently provided credentials. This is only one half of the information needed to make access control decisions. In chapter 5 we look into the other half: identification of a user based on the credentials he provides.

5

Instance identification

We may have to deal with incomplete information about a person who wants to access the system. For example, when a person calls, we at first only have his telephone number. This incomplete information may be sufficient to positively identify the person, allowing him to access products while easing the authentication process. For some products, we may not even need a completely positive identification. How can we have as much confidence in the identity of the person as needed, while burdening him as little as possible with questions to verify his identity?

Data integration is concerned with correctly identifying records that refer to the same real-world entity. This process is usually called entity resolution. This knowledge can be used to match records from two distinct tables, allowing their integration into one table [24]. Many techniques and methods have been developed for data integration over the last decades. Elmagarmid et al. [24] and Halevy et al. [30] give an overview of the field.

Our problem is a simplification of entity resolution. We only have one tuple to match against a data source, instead of two data sources that need to be matched against each other. Data cleaning is not an issue, since we ask the user to enter a certain piece of information. We know what field to compare the user's input with, because we have asked him to enter that specific field in the first place. Since we want to identify users, we will refer to this process as *instance identification*.

5.1 Duplicate detection

Duplicate detection is a form of classification problem. Suppose we have two tables, A and B , that we want to match. Each tuple pair $\langle a, b \rangle$ ($a \in A, b \in B$) will be compared and assigned to either the class M , for “matched”, or U , for “unmatched” [26]. Newcombe et al. [40] and Fellegi and Sunter [26] are the founders of the

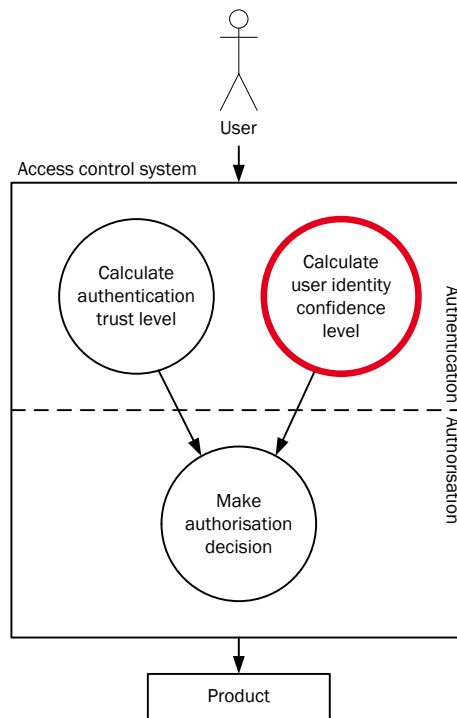


Figure 5.1: Schematic layout of the dynamic access control system

field of duplicate detection. They were the first to present and refine theories for this.

Elmagarmid et al. [24] describe the notation that is currently used in the field. Each tuple $\langle a, b \rangle$ is represented as a random vector $\vec{x} = [x_1, \dots, x_n]^T$. The values of x_i represent the amount of equality for the i th field of records a and b , that exists in both records. This can be a binary, for example, 0 for no match and 1 for a match, or a real number, corresponding to the amount of similarity of the fields. The variable n , or $|\vec{x}|$, corresponds to the number of comparable fields of A and B .

For most duplicate detection mechanisms, the matching score is not represented by a binary (match or no match) but by a real number. There are several methods to obtain the value of this number. Usually, a string matching method is used to determine the matching score.

We, however, assume that information in the database and information entered by the user are both correct. The amount of similarity per field will therefore be a binary. The information either matches or it does not.

5.2 Identifying records

Suppose we have the database as shown in table 5.1.

The system receives an incoming call from telephone number 1234. Based on

Table 5.1: Database contents

First name	Last name	Tel.no.	ID	Address	Postal code	City
Alice	Smith	1234	1	1 High St.	1000	Mytown
Bob	Anderson	1235	2	9 High St.	1001	Mytown
Charlie	Anderson	1234	3	1 High St.	1000	Mytown
Dave	Robertson	1236	4	7 High St.	1000	Mytown
Elisa	Peterson	1235	5	9 High St.	1001	Mytown

this information, it decides that either Alice or Charlie is on the telephone. Extra information is needed for the requested service, so the system asks the caller to enter his last name.¹ The caller enters “Anderson”. There are now three records that match on either or both of the fields, as can be seen in table 5.2.

Table 5.2: Instance identification for this database

First name	Last name	Tel.no.	ID	Address	Postal code	City
Alice	Smith	1234	1	1 High St.	1000	Mytown
Bob	Anderson	1235	2	9 High St.	1001	Mytown
Charlie	Anderson	1234	3	1 High St.	1000	Mytown
Dave	Robertson	1236	4	7 High St.	1000	Mytown
Elisa	Peterson	1235	5	9 High St.	1001	Mytown

The system could decide that it knows for certain that Charlie is on the telephone. But what if Bob is calling with his brother’s telephone? Or Alice and Charlie are married and Alice is calling, and now uses her husband’s last name instead of her own? We need to take these options into account.

In this case, we can easily ascertain who is calling by asking the ID. But the ID may not be known to the user, or may be really difficult for him to retrieve. The system needs to be able to reason on this as well. Therefore we introduce the concepts of *commonness* and *effort*.

The system will, in each step, aim to obtain a credential that is as unique as possible while also easy for the user to provide. This will usually be a trade-off. The most unique credential is some kind of ID which may only be present on legal documents. Users usually do not know such an ID by heart. Things they do know by heart, for example, postal code or date of birth, are not as unique.

We discuss how to assess how unique a value for a credential is in section 5.2.1. We do not elaborate on effort in this chapter, but in section 6.2.2 instead. The reason for this is that the amount of effort needed to provide a credential has no influence on instance identification, but is only used for deciding which credential to ask next.

¹How the system decides which questions to ask is treated in chapter 6.

5.2.1 Commonness

We want the user to provide credentials of which the content is not common. A last name like Smith is very common and will yield little extra confidence in the identity of the user; however, when a user has an exotic last name, this will be more unique and we will gain more confidence in his identity. The system will have to ask the user for the credential that has many unique values and not many common ones.

Definition 8 (Commonness) *A factor that indicates how common a certain value for a certain credential is.*

We will define a function for commonness, v , from Latin *vulgaris*, meaning “common” or “ordinary”, so that the system can objectively reason on it. In this regard, we define n to be the total number of records and m to be the number of matches, that is, records that contain the same value x for a credential a .

$$v_{a \leftarrow x} = \frac{m}{n} \quad (5.1)$$

In table 5.1, the commonness of “Anderson” for the credential “last name” is $\frac{2}{5}$. The commonness of “Mytown” for the credential “city” is 1, which means that all records in the table share this value for this credential and we would learn nothing by asking it.

After each information request iteration, there is one credential that is the most unique for the current data set. The system would like to know this so it can rule out the most users. To know which credential this is, we can calculate the average commonness for each credential.

The average commonness for “last name” in table 5.1 is the following.

$$\frac{\frac{2}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5}}{4} = \frac{1}{4}$$

We can see that the numerator in this fraction always adds up to 1. Therefore, the only information we need is how many distinct groups exist for this credential. We call this number of distinct groups j . The equation for average commonness is shown in equation 5.2.

$$\bar{v}_a = \frac{1}{j} \quad (5.2)$$

When we normalise this number to the number of records in the table, by multiplying with n , we get the average number of occurrences for any value in the table. A last name occurs on average $\frac{1}{4} \cdot 5 = 1.25$ times. We do not use this normalisation in our calculations, because a value that occurs on average 1.25 times in a table of 500 records is far less common than when it occurs that many times in a table of 5 records.

The system will calculate the commonness based only on the records that are left when a selection has been made with the credentials that are currently known. This is done because the goal is to eliminate as many possible users as quickly as possible. We therefore need to know the credential that has the lowest average commonness for the current selection of users that may still be the actual user. This value may differ from the lowest average commonness of the entire table.

Table 5.3 shows an excerpt of table 5.2 containing only the records that are left after selecting on telephone number. The average commonness for address, postal code and city all equal 1, while the average commonness for first name, last name and ID all equal $\frac{1}{2}$.

Table 5.3: Excerpt of identified records

First name	Last name	Tel.no.	ID	Address	Postal code	City
Alice	Smith	1234	1	1 High St.	1000	Mytown
Charlie	Anderson	1234	3	1 High St.	1000	Mytown

In table 5.1 the address has an average commonness of $\frac{1}{3}$. Because an address is fairly easy for the user to provide, the system may decide to ask for this, based on the average commonness of the entire table. But, as we can see in table 5.3, the address is the same for both records. It makes no sense to ask this information. Because of this, we base decisions on the lowest average commonness of the current selection.

5.2.2 Calculating confidence

When asking credentials, we may find multiple users that match. The probability that a matched user is actually the user that wants access, is not the same for every match. A naive solution would be to divide the probabilities equally over all possible values; however, this rarely reflects reality.

We can see in table 5.2 that Charlie matches on two credentials, while Alice and Bob both only match on one credential. The probability that Charlie is trying to access the system is therefore higher than the probability that either Alice or Bob is trying to gain access. This needs to be reflected in the confidence score.

van Keulen et al. [56] use a simple method to calculate the confidence score in their data integration solution. They integrate data from multiple devices. They increase a possible value's confidence score by one every time a device claims that that possibility is true.

We could adapt this method to increase the confidence score by one for each credential that is a match for a user, but this does not reflect that the confidence in the user's identity should increase more when he supplies a credential that is less common. The confidence should increase more when a user provides a username than when he provides the name of the city he lives in.

We have already defined the concept of commonness. A credential for which more unique values exist, that thus has a lower commonness value, should provide more confidence in the identity of user u . The confidence value we give to a single credential, when it is a match for user u , is the following.

$$c_a(u) = (1 - \bar{v}_a) \quad (5.3)$$

When a credential has the same value for all users in the database, its average commonness $\bar{v} = 1$. We can see that the confidence in the identity of a user

supplying such a credential is zero. This is wanted behaviour, since we have no new information that could confirm the user's identity.

When users provide multiple credentials we simply add the confidence scores.

$$c(u) = c_{a_1}(u) + c_{a_2}(u) + \dots + c_{a_x}(u) \quad (5.4)$$

We could normalise this into a probability space, where the confidence scores of all users add up to one, but then we lose the information about the fact that every credential provides additional confidence. When normalising into a probability space, the only thing that may happen when an extra credential is added, is that the probabilities shift between users. We cannot deduce whether a user deserves additional confidence because he has entered additional credentials.

There is one situation where the normalisation into a probability space is useful. When two results have almost the same confidence scores, we can be less certain about the identity of the user than when the confidence scores differ a lot. Both scores, however, may be sufficient for access to a certain product. This means the system will grant access, because it thinks it knows who wants to log in, but it may be wrong.

When using a probability space, this problem cannot occur. When only two results have a high, similar confidence score, and the scores of other results are negligible, they will both have a probability of about 50%. The system may then, for example, require a probability of at least 75% to prevent it from selecting the wrong result. This, however, has to be done in combination with the regular confidence score, because of the flaw of a probability space that we have mentioned earlier. We do not know how many credentials a user has provided solely from the probability.

Example 5.1 *A user is calling using telephone number 1234 and identified himself using his last name, "Anderson". What are the confidences for the users in table 5.1?*

We can see the resulting confidences in table 5.4. Observe that Bob receives a higher confidence score than Alice. This is because "last name" has a lower commonness value than "telephone number".

Table 5.4: Instance identification with confidences

First name	Last name	Tel.no.	ID	Address	Post.c.	City	c
Alice	Smith	1234	1	1 High St.	1000	Mytown	0.67
Bob	Anderson	1235	2	9 High St.	1001	Mytown	0.75
Charlie	Anderson	1234	3	1 High St.	1000	Mytown	1.42
Dave	Robertson	1236	4	7 High St.	1000	Mytown	0
Elisa	Peterson	1235	5	9 High St.	1001	Mytown	0

5.2.3 Unknown records

Suppose Fred Anderson intends to live with his brother Charlie. Since he does not live there yet, his record has not yet been entered into the system. When asked for his last name and telephone number, he would enter "Anderson" and 1234, since

that will be his telephone number when he is living with Charlie. As we can see in table 5.4, the system would incorrectly identify Fred as Charlie.

Fred is not the only one who may be incorrectly identified. When accessing a product that only requires more or less publicly available information for its security, an attacker may also on purpose try to impersonate someone else.

Whether this is a problem or not differs per product. The needed security level and confidence score need to be tweaked per product to ensure that no faulty identifications can occur for products for which this is necessary. A higher security level and confidence score will force the system to choose credentials that are really unique, and known only to their rightful owner. The system knows that a credential a is unique when $\bar{v}_a = \frac{1}{n}$. We will treat this in more detail in chapter 6.

5.3 String matching

We assume that both the information that resides in the database and the credentials that are entered by the user are correct. Therefore we do not need string matching techniques, like in regular entity resolution. Nevertheless, there are some cases where string matching may increase the efficiency of the system.

When we receive an e-mail from an e-mail address that is not in our system, we can use string matching to tie this address to a user's name. E-mail addresses regularly look like *firstname.lastname@example.com* or *a.b.lastname@example.com*. We can try to match these e-mail addresses to first and last names that already are in the database.

5.4 Conclusion

In this chapter we have defined functions to determine how common the values of a credential are on average. This can be used by the system to decide which credential to ask next, making the probability that an additional credential is useful to provide additional confidence in the identity of a user as high as possible.

Furthermore, we have defined how to calculate the confidence score for users based on the credentials they have provided. The system uses this to identify who is trying to gain access and whether we have enough confidence in the user's identity to grant him access.

In chapter 4 we have seen how we can assess the amount of security provided by credentials. Chapter 5 elaborated on how to identify a user based on the credentials he provided. In chapter 6, we show how the system makes its decisions based on the information it has obtained using the assessments from chapters 4 and 5.

6

Making decisions

In chapters 4 and 5 we have defined two concepts that the dynamic access control system will use to decide whether to grant a user access or not. This chapter will elaborate on how these concepts will be used to make this decision. How this fits in the system can be seen in figure 6.1.

The decision-making process consists of two distinct steps. The first step decides whether to allow access, deny access or ask an additional question. The second step only exists when the system decides that an additional question is needed. It then needs to decide which question to ask. In the following sections we will discuss these two steps in detail.

6.1 Deciding on allowing access

As we have seen in section 2.2, authorisation decisions are usually the outcome of a function that maps input parameters to either allowing or denying access. Recall this definition from equation 2.1:

$$\delta : \text{subject} \times \text{operation} \times \text{object} \rightarrow \{\text{allow}, \text{deny}\}$$

Our system will also use such an authorisation function; however, we have other input parameters and other possible output values. The dynamic access control system does not only allow or deny access, but can also ask additional credentials to the user.

Our access control function is shown below. The outcome of this function is the result of the first step. When the outcome is either allow or deny, the process finishes.

$$\delta : \text{known representation} \times \text{channel} \times \text{product} \rightarrow \{\text{allow}, \text{ask}(\text{credential}), \text{deny}\} \quad (6.1)$$

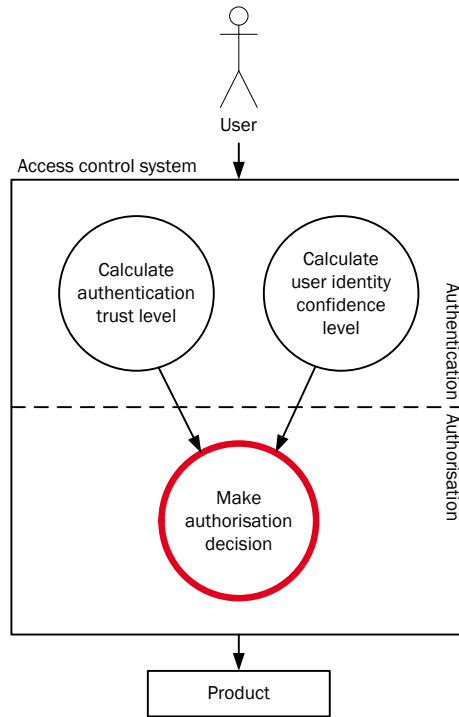


Figure 6.1: Schematic layout of the dynamic access control system

This decision can be made based on a number of criteria. Table 6.1 shows the variables that are available for our system to make a decision. These variables have all been introduced in chapter 3. Some of these variables are input by the user or other systems, others need to be calculated.

Table 6.1: List of variables

Symbol	Description
R^K	Known representation
A^K	Known credentials
χ	Used communication channel
p	Requested product
l	Authentication trust level (ch. 4)
\check{R}	Identities with confidences (ch. 5)
A^N	Needed credentials
l^N	Needed authentication trust level
c^N	Needed identity confidence

The decision to allow access ultimately only depends on the security level l and the identity confidences in \check{R} . When these are both higher than the required values, defined in l^N and c^N for each product, the system will grant access.

How high these two values need to be exactly differs per system and per product. We will define these values for the products we will use in our prototype.

When allowing access is not an option, because there is not enough security or identity confidence yet, the system can choose between completely denying access, or asking additional questions. The system will deny access when there are no more credentials that the system can ask using communication channel χ . If there are additional credentials, the system will ask for them first.

6.2 Asking for additional credentials

When the system has decided that it should ask for another credential, which one should it ask? This decision is based on a number of factors. Getting a security level and identity confidence that is high enough is the first priority of the system. In the previous chapters we have seen how the dynamic access control system determines the security level and identity confidence.

There is a number of other factors the system also takes into account. The system can, of course, only ask for credentials that are available for the used communication channel. Furthermore, it looks at the amount of effort needed for a user to provide a credential. These concepts are discussed in the next sections.

6.2.1 Available credentials

Not all credentials can be used with all communication channels. The system needs to know what credentials it can ask via the communication channel that the user is using. We could store, for each credential, the communication channels that this credential can be used with, but this would be a lot of work. It would also require extensive maintenance once a new credential or communication channel is introduced.

We will instead divide the credentials into several types. These types are shown in table 6.2. With the registration of each credential we need to include this type so that the dynamic access control system can deduce what channels it can be used for. The system can then choose the appropriate credentials for the communication channel that is used.

Table 6.2: Types of input

Type of input	Examples
Numeric	PIN, telephone number
Alphabetic	last name, city
Alphanumeric	username, postal code
ASCII printable characters	password
Binary data	biometric

A telephone supports numeric credentials, so it can be used to enter credentials that belong to this group. A system that supports alphanumeric credentials obviously also supports numeric and alphabetic credentials. We can group the types of credentials in this way to further ease the categorisation of communication channels.

Table 6.3: List of characteristics

Characteristic	
A	Personal information
B	Used at least once a week
C	Will never change
D	Self-made
E	Shorter than 10 characters or pronounceable
F	No additional equipment needed
G	Easily reachable

6.2.2 Effort

The system may have decided on a credential it wants to know to ascertain the identity of the user, but this credential may be nearly impossible for the user to provide. In this case it would be better to have the system ask other, probably more, credentials that are easier for the user to give. For this, we introduce the concept of *effort*.

Definition 9 (Effort) *A factor that indicates how difficult it is for the user, or how much trouble he has to go through, to provide a credential.*

We will objectively give credentials an effort score based on a number of characteristics each credential either has or not. Table 6.3 contains these characteristics. An ideal credential in terms of effort would have all these characteristics. For each characteristic a credential does *not* have, the effort score is increased by 1.

For example, the only applicable characteristic for a last name is that it is not self-made, so it has an effort score of 1. A randomly generated password for a website that the user visits regularly would not have characteristics A, C and D and would therefore have an effort score of 3.

6.2.3 Deciding what to ask

We ask additional credentials to enlarge the level of security and our confidence in the identity of the user. Each decision on which credential to ask needs to be a consideration based on the increase in security and confidence that is needed, and the amount of effort it takes for the user to provide the credential.

An important observation is that we only want the security level and the amount of confidence to be sufficient for the requested product. If we were to pursue the highest security possible, the system would ask everyone for an iris scan, fingerprint scan and voice recognition, all the time. We want to increase the security level so that it reaches the level that is needed for the requested product, and does not go over it. This of itself already decreases the amount of effort that is needed to supply credentials.

The influence that effort has on the selection of credentials differs per product. For products that need higher security, keeping the effort score low is less important than for products that only need low security. When multiple credentials give an

equal increase in security and confidence, the system chooses the credential that needs the smallest amount of effort.

6.3 Assessing the needed security level and identity confidence

There are two approaches to determining what security level and identity confidence are actually needed for a product. One approach does not look at the product itself, but at the data it reads or alters to determine the needed security level and identity confidence. The second one looks at the preferred credentials for each product.

The used data method works better when there are a large number of products that need a security level, while the preferred credentials method is better suited for a smaller number of products. Of course it is also possible to combine these methods, by creating a list of preferred credentials based on the information that products use.

6.3.1 Used data

There can be an unlimited number of products that need assessment for their needed security level. Assessing this separately for each product can take ages. This is a more general approach.

First of all, we do not allocate a unique needed security level and identity confidence to each product; instead, we divide the products into a number of groups. To ease the assignment of products to groups, we assess their needed security level and identity confidence based on the information that they read or alter. Usage of more sensitive information requires a higher security level and a higher confidence in the user's identity.

For example, we can define a security level and identity confidence for an application that writes to the database that contains reports on the environment, like broken lamp posts. These values probably are lower than those for an application that reads personal information of the user.

What these values are exactly differs per application. An expert in the field needs to determine these.

6.3.2 Preferred credentials

When using this approach, we determine what credentials we would like to see used for a certain product. We can calculate the security level that these credentials deliver, and estimate the identity confidence that we will get from using these credentials. These values are entered for the product. The system will then choose the preferred path or a similar group of credentials.

When we would like to see that the system chooses to ask for a username and password to obtain sufficient security, we can calculate what the security level

is when these two credentials are combined. Based on the current values in the database, we can also make an approximation of the identity confidence these values will deliver when entered correctly. When we then enter these values, the system will either choose these credentials or similar ones that deliver the same amount of security.

6.4 Conclusion

In this chapter we have seen how the system decides on granting or denying a user access. We have also shown how the needed security level and identity confidence can be determined for a product. This concludes our coverage of the internals of the system.

7

Evaluation

Our goal for the evaluation of the system is twofold. We want to know whether our system can make the correct decisions when used, but we also want to know how users perceive the system's security and ease of use. It is important that users do not see the system as insecure or difficult if it is to be used frequently.

To evaluate our ideas we have built a prototype of a dynamic authentication system. A group of users has tested this prototype. They were given directions to log into the system using a set of credentials that was given to them. The system logged the amount of time this took. The participants also filled in a questionnaire. This chapter explains the process of the evaluation.

7.1 General set-up

The participants first received a piece of paper that contained instructions for the evaluation. These instructions are shown in appendix A. Each instruction form received a unique number. The numbers were not consecutive to avoid typing errors. The numbers on the forms increased in steps of 3.

They were asked to try to log into the system using credentials from a fictitious user. These credentials were also on the paper with the instructions. They were similar to the one shown in figure 7.1.

The fictitious users have been chosen so that they share values for some of the credentials. The system contains:

- users sharing a last name;
- users living in the same house, sharing postal code, house number and telephone number;
- users living in the same street, sharing postal code;

- users with similar telephone numbers;
- users born in the same municipality.

Once the desired security level has been reached, the prototype will choose the credential that has the highest discriminatory value to reach the needed identity confidence as quickly as possible.

We have chosen a set of credentials that allows the system to choose a number of different paths to achieve enough security. There is a large number of credentials that offer low security, and less credentials that offer higher security. The system can choose whether it uses one credential that offers higher security, or a number of credentials that offer lower security. The complete list of credentials including their security level is included in appendix C.

First name	Jan
Last name	Meerwijk
Postal code	9639
House number	13
Municipality of birth	Berkensveen
Telephone number	119452
Passport number	5695233
Citizen ID	10038596
Username	jmeerwijk
Password	hYe3EVE4
Access code	3942

Figure 7.1: Example card

The paper with instructions told the participants which products they needed to log into. The participants tried to gain access to two products, the latter always needing a higher security level and identity confidence than the former. This is to test their reaction to additional questions when requesting a product that requires a higher security level and identity confidence.

The set of products has been chosen so that the products can be logged into using a mix of credentials. The needed security levels of these products have been adapted to the available credentials. One product needs a low security level that can be reached by asking two credentials that offer low security, while another product needs a high security level that can only be obtained by asking a credential that offers this security level or multiple credentials that can together offer this security level. The exact numbers that were used are shown in table 7.1.

Table 7.1: Specifications of used products

Product name	Needed identity confidence	Needed security level
Report broken lamp post	0.7	0.050
Make appointment	1	0.301
Request certificate of residence	2	1.046

The next step for participants is to actually log into the system. During this process, the prototype registers what the participants enter, what the current levels for probability of discovery and identity confidence are and what decisions the system makes based on this. It also logs whether the login procedure succeeds or fails in the end. More information on the prototype is given in section 7.2.

After they finished using the dynamic access control system, participants filled in a questionnaire to gauge their opinion on the system. The goals of the questionnaire and how this questionnaire was created are treated in section 7.3.

7.2 Prototype

Our prototype has been built to run in a web browser, just like regular access control systems on the web. The first screen shows a short explanation of the evaluation and invites the user to start using the prototype, as can be seen in figure 7.2. On the second screen, the participant selects the product he wants access to. This screen is shown in figure 7.3.

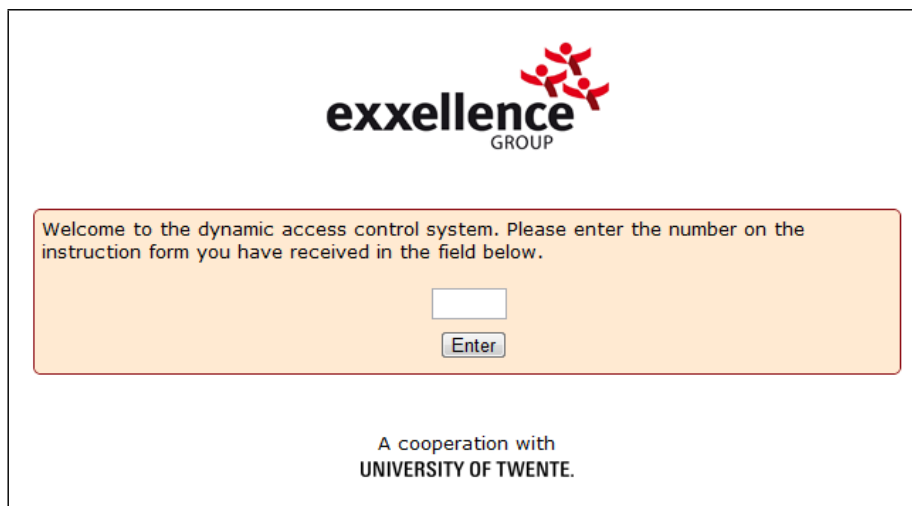


Figure 7.2: Enter instruction form number

After selection of the product, the authentication process starts. The system asks a number of questions. Once the probability of compromise is low enough and the identity confidence is high enough, the system will grant the participant access. If there are no more questions to ask, and there is not enough security or identity confidence yet, access will be denied. A screen that asks the user for a credential is shown in figure 7.4.

While the participant is attempting to gain access, the prototype logs different aspects of the login process. First of all, it stores the requested product. For each step it stores the current probability of compromise, the current identity confidence for each user that it has found, and the requested credential. It also stores the time it took to complete a step.

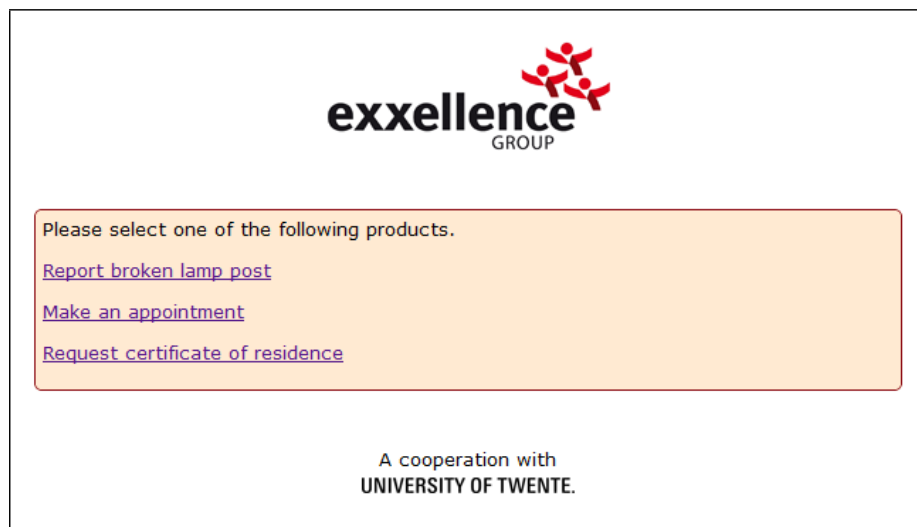


Figure 7.3: Select a product

7.2.1 Implemented functionality

The prototype implements a large part of the methods we have described in previous chapters. It finds the most suitable credential to ask based on probability of compromise and identity confidence.

It does not take effort into account when deciding on which credential to ask. This prototype uses only a small set of credentials that do not differ a lot in the amount of effort that is required to obtain them. Uncommon credentials or credentials that require a lot of effort to obtain, like fingerprint recognition and one-time password generators, are not present in this evaluation. Furthermore, the participants received all values for their credentials on a card. While the system may then favour credentials that require less effort, the effect on the evaluation is minimal.

Since the credentials used are similar, the prototype does also not take the similarity coefficient into account when calculating the new probability of compromise.

The prototype also does not take into account that some credentials may be required for access to a certain product. It only checks the probability of compromise and identity confidence to deny or allow access. Since this is not an actual production environment, we found it more important to test whether our model works and how users perceive it than to mimic a production environment as accurately as possible. The prototype is used to evaluate the access control mechanism and not the business process that belongs to the product.

7.3 Questionnaire

The questions we would like to see answered are the following.

excellence
GROUP

You have selected "Request certificate of residence". To identify you for access to this product, we request that you enter your **password** below.

A cooperation with
UNIVERSITY OF TWENTE.

Figure 7.4: Ask for a credential

1. How do users experience the ease of use when using the dynamic access control system?
2. How do users experience the amount of safety when using the dynamic access control system?
3. Does the amount of computer experience influence these experiences?

The variables in this questionnaire are *computer experience*, *ease of use* and *amount of security*. We want to compare groups of users that differ in the amount of computer experience. We can divide this group into five categories:

- Very little experience
- Little experience
- Neither little nor much experience
- Much experience
- Very much experience

We are interested in whether our system is useful for everyone, regardless of the amount of computer experience he or she has. The amount of computer experience may also be related to other variables, for example, whether experienced computer users write passwords down more or less often.

We need enough participants for each category to be able to discern possible differences. About 10 participants per category is sufficient to be able to determine whether a connection exists between different variables; however, to be able to tell how strong this connection is, about two or three times more participants are needed [18, pg. 11].

Since we are dealing with feelings or experiences, our variables are not directly quantifiable. An accurate measurement on the strength of the connection is therefore not needed. We believe that 20 participants per category is sufficient, which means

that a total of 100 participants is adequate for this questionnaire, provided that they are equally distributed over all categories.

Within the time limits set for this research, we were unable to reach 100 participants. A total of 60 people have participated in the evaluation of the dynamic access control system, 40 males and 20 females. Section 7.4.1 investigates the characteristics of these participants.

We want to conduct this evaluation with people who may use the dynamic access control system in the future. Since this research is mainly aimed at governmental services, the system may have to be used by anyone who chooses to do business with their government online. The target population for this evaluation therefore is everyone who has access to the internet from his home.

The questions that have been asked in the questionnaire can be found in appendix B.

7.4 Results

To check for statistically significant relationships, we have created a crosstable with all variables. This crosstable shows where interesting relationships occur. We will discuss the relationships we have found in this section.

7.4.1 Demographics

Figures 7.5 and 7.6 show the age, gender and education of all participants. The education is according to the Dutch educational system. MBO stands for Middelbaar Beroepsonderwijs (vocational education), HBO stands for Hoger Beroepsonderwijs (professional education).

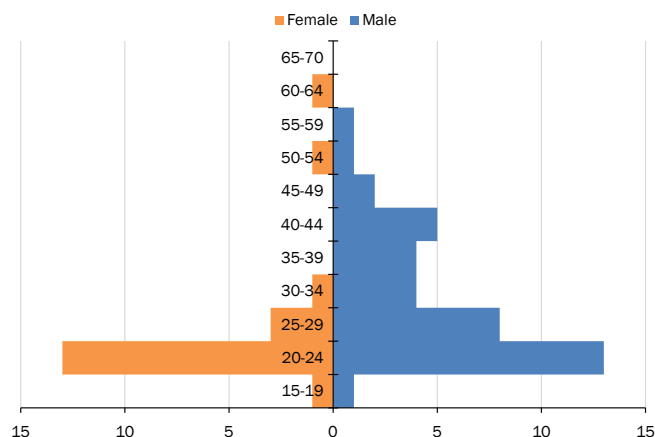


Figure 7.5: Age and gender of participants

All participants in the evaluation had above-average experience with computers, as we can see in figure 7.7. The amount of computer experience has a weak correlation

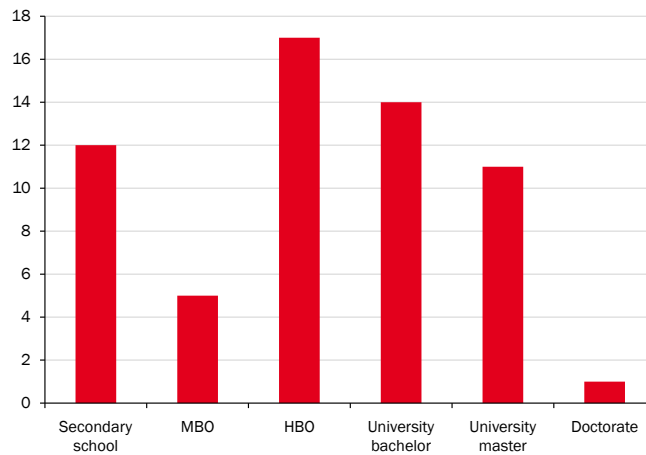


Figure 7.6: Highest completed education of participants

with how long ago participants used a computer for the first time ($N = 60, r = 0.259, p < 0.05$) and with age ($N = 59, r = 0.261, p < 0.05$).

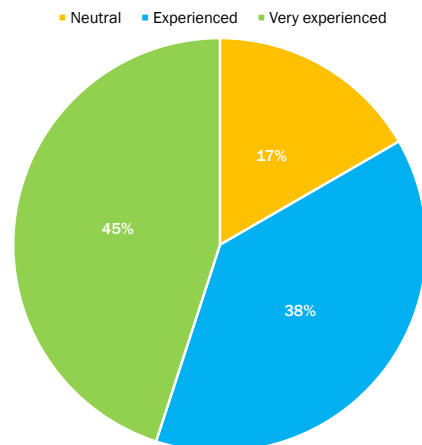


Figure 7.7: Amount of computer experience

7.4.2 Prerequisites

We have chosen three authentication methods to compare the dynamic system with. These three methods were chosen to represent low, medium and high security levels. That we have mainly succeeded with this can be seen in figure 7.8. One interesting observation is that people perceive authentication based on caller ID to be more secure than we had anticipated. We need to take this into account when we compare caller ID with our dynamic access control system. The detailed results can be found in appendix E.1.

A number of participants remarked that caller ID is never the only credential used for authentication. When they call to a helpdesk, the agent asks control questions to ascertain their identity, like date of birth or customer number. Some participants may have assumed that this is also the case for the caller ID used in our research and may therefore believe this system is more secure than it was intended to be.

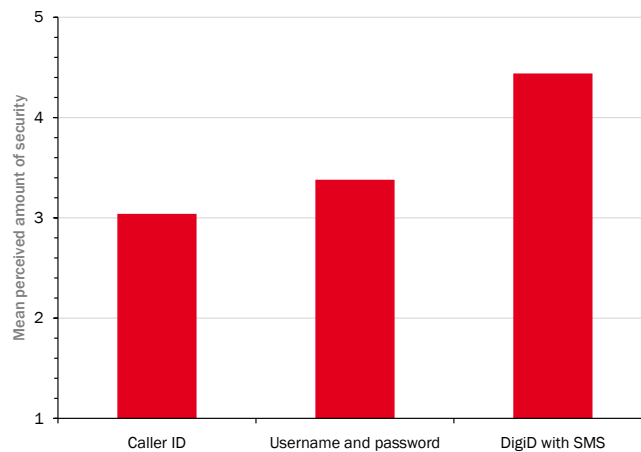


Figure 7.8: Mean perceived security per authentication method (1: very insecure, 5: very secure)

We have also chosen three products to represent products that need a low, medium or high security level. We have not directly asked people what security level they perceive to be needed for each product. It is difficult to quantify this and interpretation of these values will differ for each participant. Instead, we have asked them whether they found the authentication methods secure enough for the products they have logged into during the evaluation. The results of this can be found in figure 7.9. The exact data can be found in appendices E.2 through E.5.

On first glance the product representation for low, medium and high security levels is correct. Note that there are two columns for product B. Participants in the evaluation had to log into a combination of two products. There were three combinations possible: AB, AC and BC. They could either log into product B first or second. A peculiar observation is that participants, who had to log into product B first, value the security of the authentication methods higher than participants who had to log into product B second.

We believe this is caused by the fact that people compare the amount of security they perceive as necessary to the products they have used. People who had to log into product B second compare the security they perceive to be necessary for this product with the security of product A, while people who had to log into product B first compare this to product C. The participants who had combination AB obviously feel that product B needs more security than product A, but they do not know about product C, so they do not know where they place product B's needed security in relation to product C's needed security. We believe the effect is similar for combination BC.

We have checked whether the measurements for products A and C also differ based

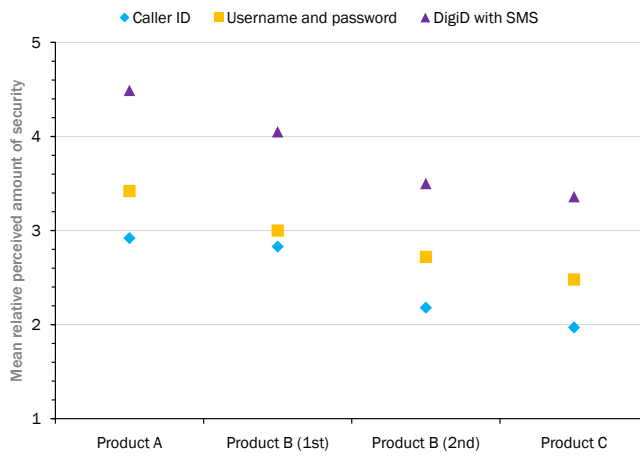


Figure 7.9: Relative security per authentication method per product (1: much too insecure, 3: exactly right, 5: much too secure)

on what the other product is. For these, there is no significant connection.

7.4.3 Reference authentication methods

We have not only measured how secure the participants found existing authentication methods, but also how difficult and how pleasant they are to use. These measurements have been used as a basis to be able to discern how difficult and how pleasant the participants believe the dynamic access control system is. The results of these questions can be seen in figure 7.10. The exact data can be found in appendices E.6 and E.7.

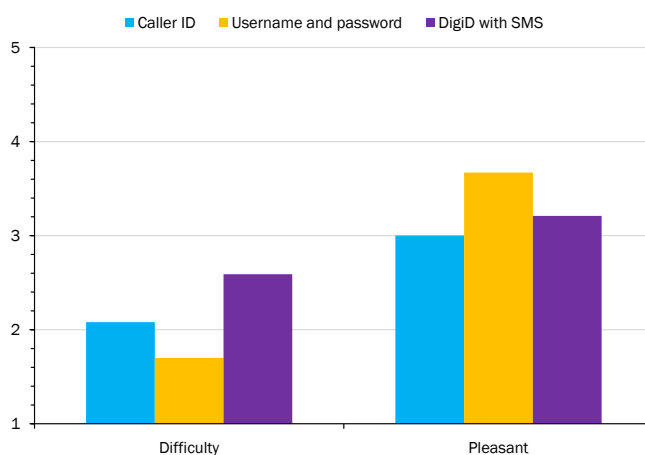


Figure 7.10: Difficulty and pleasant (1: very easy, 5: very difficult; 1: very unpleasant, 5: very pleasant)

We can see that the participants find the username and password the least difficult and the most pleasant system to use. This probably is due to the fact that this system is so widespread. Most participants therefore have used this system a lot. Although the system using caller ID requires no intervention from the user, people still find this system more difficult to use than username and password. This may indicate a flaw in the evaluation. The system using caller ID was devised to be the simplest system in the test, needing no input from the user for access control. Our assumption was that the participants would find this system not difficult at all. Apparently, the participants did not see it this way. This may have to do with the assumption that control questions will be asked when using caller ID, as we have mentioned before. With these control questions, this authentication method becomes less pleasant and more difficult to use.

7.4.4 DACS

Figure 7.11 shows that the system did a good job of adapting itself to the product that was requested. In general, the participants found the dynamic access control system to be a little too insecure, but the relative amount of security stays more or less the same for all products. Achieving a higher average security can be reached by tweaking the configuration of the system. The exact data can be found in appendix E.8.

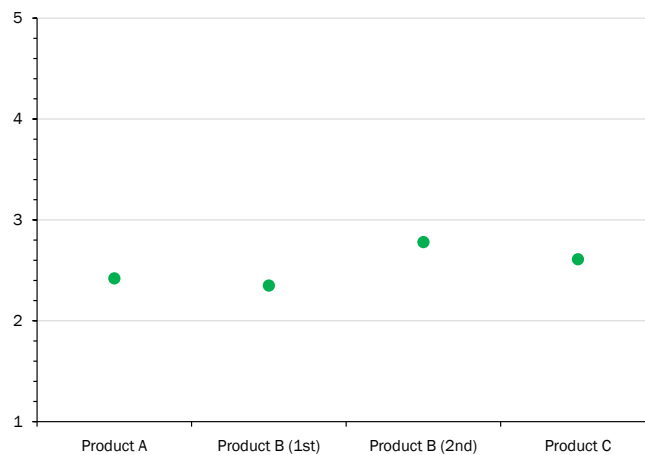


Figure 7.11: Relative security per product for DACS (1: much too insecure, 3: exactly right, 5: much too secure)

The questionnaire also asked people how they experienced working with the system: did they find it annoying that the questions that they needed to answer were not known beforehand? Did they find it annoying that they had to answer additional questions to gain access to a product that needs a higher security level?

Figure 7.12 shows that people are either annoyed or not annoyed by the fact that they do not know beforehand which questions they need to answer. They do not feel very strongly about it, but they are not neutral either. We have investigated

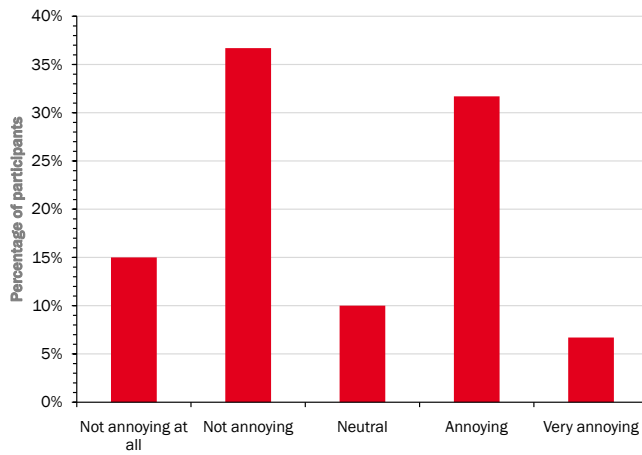


Figure 7.12: Annoyance caused by the fact that questions are not known beforehand

whether one group of users finds this annoying while another group of users finds this not annoying, but we have found no significant connections for this.

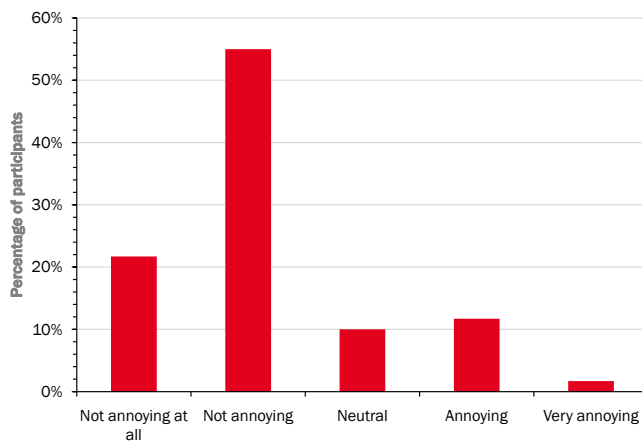


Figure 7.13: Annoyance caused by asking additional questions between requests, for higher security

The participants in the evaluation do not find it annoying that they have to answer additional questions when higher security is needed for the second product they request, as figure 7.13 shows. It appears that security is of a higher concern to these users than usability.

7.4.5 Comparing DACS to existing access control methods

We have asked what people think of the dynamic access control system compared to systems they already know. Do they think the dynamic system is more pleasant and easier to use or not?

The bars in figure 7.14 show how pleasant people find the three reference systems to use. The squares show how pleasant they found DACS to use with regard to the reference system. To calculate this value we have used the following formula: $\text{value} = \text{score of reference access control system} + (\text{score} - 3)$. Table 7.2 shows the scores and the adaptations to the end value they deliver for how pleasant the participants thought the system was. The table for difficulty is similar. The actual values do not have to be integers because we work with means of scores filled in by all participants. The exact results are in appendix E.9.

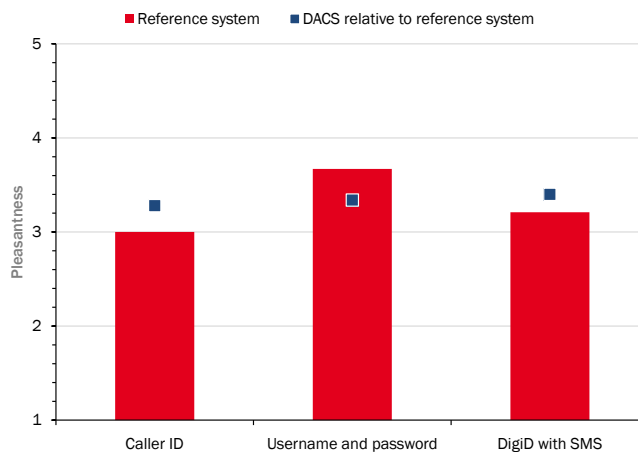


Figure 7.14: Relative perceived pleasantness for DACS

Value	Score	Adaptation
Much more pleasant	5	+2
More pleasant	4	+1
Just as pleasant	3	0
More unpleasant	2	-1
Much more unpleasant	1	-2

Table 7.2: Relative scoring

As we can see, the participants do not find DACS a lot more or less pleasant to work with than the existing systems. The combination of username and password is clearly what people are used to and what they find the most pleasant system to use. They also perceive the combination of username and password to be the least difficult to use, as can be seen in figure 7.15. They find DACS easier than both identification based on caller ID and DigiD with SMS. The exact data can be found in appendix E.10.

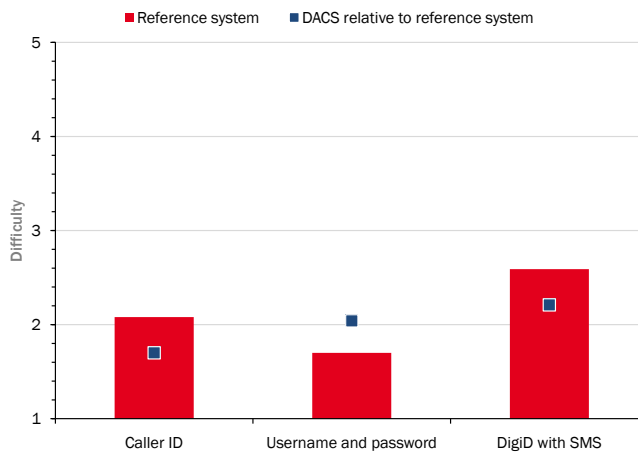


Figure 7.15: Relative perceived difficulty for DACS

Since one of our intentions was to make an access control system that would be more pleasant and easier to use, it is interesting to find out why people still value username and password higher in this regard, because this is not what we expected.

The answers from the questionnaire reveal two significant correlations. People who find it annoying that they do not know in advance which questions they need to answer find DACS to be less simple than username and password ($N = 59, r = 0.332, p < 0.01$) and caller ID ($N = 53, r = 0.336, p < 0.01$). As one participant remarked: “The main disadvantage is that you have no idea how long the login process will take.”

More participants left remarks that shed light on why they still perceive username and password to be the most pleasant and least difficult access control system to use. Multiple participants remarked that a number of “difficult” credentials was used by the dynamic access control system; information that usually is not known by heart, and information that is hard to remember.

For a number of participants it was not clear that the additional questions they needed to answer, to access a product with higher security requirements, were there to ensure this higher security, and that the system does not need to ask everything again because it has remembered the answers from the first product. These people thought that the login process started from the beginning again when they selected the second product, and that the system just asked different credentials this time. They may not have noticed the differing security requirements. Using this train of thought, it of course is more annoying to use the dynamic system, when apparently every product needs a different piece of information from the user.

Another participant assumed that it is impossible to choose your own password when using the dynamic access control system, since we compare the dynamic system with a system that does allow you to choose your own username and password. He therefore found DACS to be less pleasant and more difficult, since randomly generated passwords are very difficult to remember. Apparently the instructions and description were not clear enough about this. Since we can use any credential with

DACS, it is also possible to use a password that the user has thought of himself. The random looking passwords for the fictitious users that were given to the participants may have strengthened this participant's assumption.

One participant confirmed the usage of a password manager. This application automatically enters usernames and passwords on websites for which these have been stored. This makes using username and password very easy and pleasant, and obviously easier and more pleasant than the dynamic system, but this also makes username and password less secure.

Another cause of annoyance may be the time it takes to login. We have registered how long the login process takes for each participants. The results are shown in the histogram in figure 7.16.

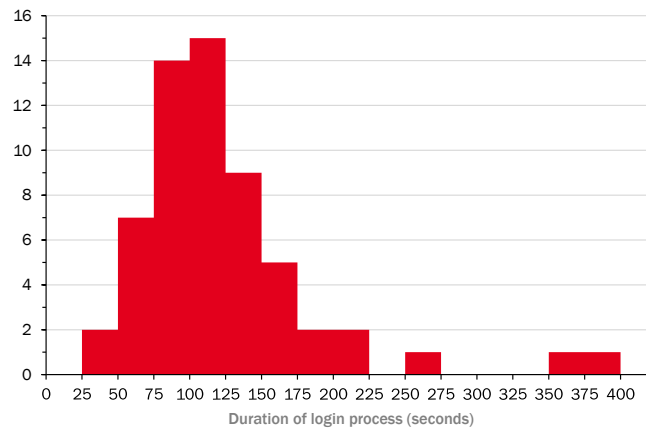


Figure 7.16: Histogram for the duration of the login process

The mean login time is 124.3 seconds, with a standard deviation of 63.0 seconds. The median login time is 109.5 seconds. Commonly used access control methods, like username and password, do usually not take around two minutes to use. Furthermore, people are not used to the dynamic system, therefore taking more time to answer the questions. These two aspects may cause participants to find the dynamic system less pleasant.

We have checked the outliers in this histogram, but nothing strange happened there. We believe these participants may have been distracted during the login process. There is no significant connection between the time that logging in takes and any other variable in the evaluation.

We expected to see correlations between the amount of computer experience and how often people write down passwords and how pleasant and easy they think DACS is. There is a weak negative correlation between the amount of computer experience and whether the participants found DACS more pleasant to use than username and password ($N = 60, r = -0.266, p < 0.05$). We believe that users with more computer experience are more used to using username and password and therefore do not find DACS as pleasant. Users with more computer experience may also more often use a password manager.

There are no other significant correlations with the amount of computer experience. However, we do not believe that this indicates that the amount of computer experience has no influence on the usability of the dynamic access control system. Participants in our research all had above-average experience with computers, which means that the dispersal of participants over the different categories is low. This makes it harder to find correlations.

Of all participants, 43.3% has written down passwords. There is no significant correlation between the amount of computer experience and having written down passwords ($N = 60, r = -0.107, p > 0.05$). People who have written down passwords do not find the authentication method of username and password significantly less pleasant than people who have not ($N = 60, r = 0.107, p > 0.05$).

7.5 Evaluation with security experts

In a conversation with security experts from the RDW, the Dutch vehicle registration organisation, we have discussed the dynamic access control system. The RDW is responsible for all vehicle registrations. Every year, they register nearly 900,000 new vehicles and process more than 6 million change of ownership transactions. They also keep track of the almost 11 million driving licences that have been issued in The Netherlands. Furthermore, they handle information on nearly 7 million periodical vehicle checks per year and also issue certificates for approval of imported vehicles and new vehicle types. This conversation was held with, in alphabetic order, Eric Algera, security manager; Gert Maneschijn, Corporate Security Officer; Bjorn van der Schaaf, IT auditor and Alfred Velthuis, consultant.

They saw possibilities for the dynamic access control system. They believed that the system could be very useful for people who have little computer experience, since filling in information like a postal code and house number would be easier for them than to remember passwords. When someone needs to identify himself on the telephone or at a counter, this person needs to answer similar questions. This means that these people already have experience with this way of authentication, making the process easier for them. This is especially useful for products that do not require a high amount of security.

They believe that the ability of the system to work with all kinds of authentication methods makes it also suitable for products that need a higher amount of security. For these products, the system would use more secure authentication methods, like DigiD, to ensure the needed amount of security.

The only problems they saw were at implementation level. The values of credentials may reside in different databases. When this data is sent back and forth, it needs to be encrypted to ensure that it will not be intercepted. Furthermore, these values are stored somewhere during execution of the dynamic access control. This storage also needs to be secure.

A practical problem originates in privacy regulations. Before municipalities are allowed to use personal information of citizens for authentication, like the dynamic access control system does, they need permission from the Dutch Data Protection

Authority. Since this research mainly looks into the principles of dynamic access control, these issues can be tackled later.

The security experts from the RDW concluded that a system like this may be very useful, but that good thought needs to be given to details pertaining to the secure implementation.

7.6 Conclusion

This chapter described the evaluation we have performed using the dynamic access control system. Participants have worked with a prototype of the system and have filled in a questionnaire.

We have seen that users feel that the dynamic access control system succeeds at adapting to the required amount of security. People do not, however, like the dynamic system as much as username and password, because they do not know how long the login process will take and because the system may ask questions to which they do not know the answer by heart. We have also seen that logging in using the dynamic system takes longer, perhaps because users are unfamiliar with the system.

Participants valued username and password as the most pleasant and least difficult system to use. We believe this is because this system is very common. We, however, do see potential for the dynamic access control system. Once users are more familiar with it, its usage may be more pleasant and less difficult. Security experts confirm that there is potential, but they see a number of implementational issues that need to be tackled before DACS can gain widespread use.

We recommend to use only a small set of credentials. This way, a user can remember which credentials he may be asked for to gain access to the system. This also ensures that they are able to learn all answers by heart, if they do not know them already. Furthermore, users can become familiar with the system more quickly.

8

Discussion

8.1 Security level for nonexistent answers

When a user enters a value for a credential that can not be found in the system, his identity confidence will not rise. Should the security level rise or not?

The calculation of the security level does not need an interaction with the user database; it does not take the values that have been entered into account, but is based solely on the types of credentials that have been provided. When we assume that there are no users with bad intentions, which is of course a very unrealistic assumption to make, but we make it nevertheless for the sake of explaining the principle, every given credential provides an amount of security, regardless of whether its value exists in the database or not. So, in essence, this is the correct behaviour, which therefore has been implemented in the prototype.

In practice, however, this could lead to the following situation. A user tries to log into the system and enters a few credentials. At one moment, the identity confidence is high enough, but the security level is not. The user can then enter random values for each credential and still receive access in the end, since the security level rises even for incorrect answers. Obviously, this is not what we want.

A situation like this can be avoided by taking great care in deciding on values for security level and identity confidence. A product should not need a high security level when it only needs low identity confidence. By giving the needed security level a value that fits with the needed identity confidence, situations like this can be avoided.

Another solution is to combine the, currently distinct, concepts of security level and identity confidence into one concept. The value of this concept will only rise when a correct answer to a credential has been given. The amount that the value rises with is then based on the combination of how common the value is, that the user entered, and the security level that can be offered by the credential.

8.2 Nonexistent users

When someone, whose information is not in the database, tries to gain access, he will be asked for all credentials the system has at its disposal, before the system will deny access. The number of credentials that the system asks may be very large, while the person will never be able to gain access. This is not very user-friendly.

At one point, it will be impossible to achieve a sufficient security level and identity confidence, even when the user answers all remaining questions correctly. When this happens, the system can stop asking questions. With enough available credentials, however, this may also take a long time.

We cannot deny access after one incorrectly entered credential. The user may have made a typing error, or the actual value may have changed, but this has not been reflected in the database yet. A user may, for example, have just gotten a new telephone number that is not in the system yet. We do not want to deny access immediately, but want to give the user a chance to recover by entering additional credentials that are correct.

There is a point where the system can assume that the user is just guessing or trying to break in, for example, when five credentials have been entered incorrectly. This can be included as an extra condition that can be used to deny access. The exact number of credentials that need to have been entered incorrectly can be varied according to the needs of the products.

8.3 Typing errors

When a user makes a typing error, he will not receive the additional identity confidence that is needed to access a product. Depending on how many credentials are available, he may even be denied access, and then has to start the entire process from the beginning. This may become very annoying. While applications using username and password for access control will also deny access when a typing error has been made, the entire process of logging in using this mechanism takes less time and recovering from a typing error is therefore less annoying.

One solution is to show all credentials that the user has entered so far, with the option to remove credentials. We can only show the credential type, not the actual value, since we want to send that over the network as little as possible, and we do not want people watching the screen to notice what the user has filled in. When a credential is removed, the system will disregard the answer the user has given for that credential, put it back in the pool of available credentials and maybe ask it again, when this is necessary to achieve the needed amount of security.

A disadvantage of this is that the user has to notice himself that he has made a typing error, because we cannot show the values that were entered. Additional research is needed to find a viable solution for this.

8.4 High security products

The ability of the dynamic access control system to adapt itself to any situation keeps the login process easy. But for some products, that require high security, we may want to restrict the adaptiveness of the system. It is undesirable to have this high security product available either through one very secure credential, or through a number of credentials that are public knowledge and, together, will reach the same security level and identity confidence.

In 2008, Sarah Palin's e-mail account could be accessed by someone who had guessed the answer to her secret question [14]. At that time, she was candidate for the position of vice-president of the United States of America. This is a good example of a case where both a secure credential, the password, and a less secure credential, the secret question, both offer access to the same product, which needs the higher security that the secure credential offers.

It is true that, in our implementation, the system will first ask the most secure credential; however, when this has been entered incorrectly, the system will then continue to look for credentials to make access possible. These credentials may not be as secure.

One solution is to allow the specification of a minimum security level per credential that the system can ask. Any credentials that cannot offer this security level of themselves will also not be picked by the system. This ensures that the system will only ask credentials with a high enough security level.

8.5 Lack of familiarity with DACS

Something that may have influenced the results of the evaluation, is that people are not familiar with the dynamic access control system. They have only used the system for a number of minutes before giving their opinion. Because the system uses a completely new way of ascertaining their identity, which is alien to them and not at all like the systems they are used to, they say that the system is less pleasant and more difficult to use.

Additional research is needed to verify whether this is still the case when the participants can use the system for a longer time period. When they get used to it, they may think differently about the system. During this evaluation, they may have believed the questions they received to be strange for gaining access. After longer usage, these questions become more common and may become less awkward.

This may also be the case with the amount of trust they have in the system. When using the prototype, the participants may not have been aware of the dynamic selection of credentials to match the amount of security needed by the product. Over time, when they use the system more often, they may see that the dynamic access control system offers higher security only when it is needed, and therefore also appreciate the dynamic amount of security more. A longer pilot project is needed to verify this.

9

Related work

Other research in the area of risk-based access control has taken slightly different directions. We can make two divisions in the area of risk-based access control: the first division is based on access for either humans or computers. We can also divide the field on access control based on either credentials or trust. Our research focuses on access control for humans based on credentials.

Many related work has already been discussed in chapter 2. There are a few research subjects that were not covered as much in that chapter. We will discuss these here.

9.1 Credential-based access control

9.1.1 For humans

We have looked into a large number of credential-based access control methods for humans in chapter 2. We will therefore not discuss these here.

RSA have developed a product called Identity Verification [47] that uses public records and commercially available databases to collect information, which is then used to verify the user's identity. Brian Knaus, RSA head of identity verification services, said: "Based on your name and postal address, we use RSA services to aggregate information that only you can answer: information that is top-of-mind for you, but is harder for others to research and guess." [25]

The starting point of this system differs from ours. Using RSA Identity Verification, the user a name and postal address, and the system then searches for additional information that it can use to verify that the user provided the correct name and postal address. Our system does not collect data from external sources, and is not only used for verification, but for the entire access control process. While both systems are useful of their own regard, they have different goals.

9.1.2 For computers

Winsborough et al. [58, 59] have done research on access control for computers based on credentials. They have a different definition for credential than we; they say that “a credential is a digitally signed assertion by the credential issuer about the credential owner.” They are unforgeable and can be verified. Note that a credential is only an assertion, so the information it can contain is not restricted to identity details. They use these credentials to assert the identity of systems when they try to access other systems.

Not every system wishes to give any credential to any other system, since credentials may contain private information. These are called sensitive credentials. Winsborough et al. have built a framework that determines how much trust is needed before a system will release its sensitive credentials. These sensitive credentials may be needed because another system may not want to grant access before it has seen them.

9.2 Trust-based access control

Trust-based access control has had a good amount of interest over the past few years. Grandison and Sloman [29] give an overview of the field as it was ten years ago. Since then, a number of researchers has been working in this research area.

9.2.1 For humans

Trust-based systems for humans are currently mostly found on auction sites, like eBay. Since buyers and sellers may live all over the world and do not know each other, there is no possibility to build up trust. These systems have each buyer and seller rate each other so that all opinions together may give an indication of how trustworthy this person is [46]. These systems are also called reputation systems. They do not enforce access control but give a quantified amount of trust so that users can gauge whether they want to do business with the seller.

Research on access control based on trust is mostly done in the area of ubiquitous computing. Giang et al. [28] have developed a framework that is able to make access control decisions based on the amount of trust that the system has in the person trying to connect. This amount of trust is based on earlier accesses to the system and on the amount of trust that peers of the system have in this person. This information, combined, gives a quantified indication of trust, which can then be used to base access control decisions on.

9.2.2 For computers

Dimmock et al. also focus on trust-based access control. Their main focus is on trust between computing entities for ubiquitous computing. In [19], Dimmock explains how he envisions that a system, that bases access control on trust, is to take its decisions. The system would derive the costs and probabilities of all possible

outcomes and choose the optimal one. With other people, Dimmock then proceeds to build a framework that can be used to have computing entities make decisions on what they allow and what not. This framework may ultimately be used for ubiquitous computing [20, 21].

Ni and Luo [41] look into two problems that have appeared with trust-based access control: how can we quantify trust and how can trust be incorporated in access control mechanisms. They define a number of trust levels, that can change based on direct trust and reputation. Direct trust is how much an entity trusts another. This amount changes with each time two entities service each other. Reputation is based on the amount of trust that other entities have with this entity. Using these two concepts, they invent equations that can be used to quantify trust. Based on these quantifications, services can either allow or disallow clients.

9.3 Human factors in access control

Our evaluation asked participants to rate the ease of use and pleasantness of a number of access control methods. More research has already been done in this subject area. Earlier in this report we have discussed research by Adams and Sasse [1], Dourish et al. [22], Gaw and Felten [27], and Schneier [52] that covers this. A complete usability study of our access control system is outside of the scope of this report. We do, however, want to mention some articles that contain research in this direction.

Zurko et al. [64] have developed an authorisation service, called Adage. The authors have placed psychological acceptability at the center of design for this service. They have built an architecture for role-based access control, and have done extensive usability testing on it. One of the tests they have performed is the following. They have divided the graphical user interface of the system into several concepts. For each concept, they asked the participant to rate the clarity of this concept on a scale from obvious to confusing. They also asked the participants to evaluate their interaction in general on a scale from very satisfying to very frustrating. These are two aspects that we have not touched in the evaluation of our system, and that may lead to interesting results.

Yee [63] discusses a set of design principles that can be used to design usable secure software. Examples of these principles are *path of least resistance*, meaning that “the most natural way to do any task should also be the most secure way” and *identifiability*, which says that “the interface should enforce that distinct objects and distinct actors have unspoofably identifiable and distinguishable representations.” The author also shows where some of these principles are violated in real-world situations, to back up the claim that “consideration of human factors is essential for security.” These design principles could be applied to any new security-related system, like the dynamic access control system, to check for its usability.

10

Conclusions

Municipalities and other governmental institutions want to offer more products online. Access control for these products is done using DigiD, which is mandatory for governmental institutions. Unfortunately, DigiD has flaws. This hinders the adoption of these products.

The first flaw is DigiD's complexity. It has been designed to provide high security. The system offers additional security measures like two-factor authentication via sms. Requesting a DigiD is done via an authentication code that is sent via regular mail, after which the user can choose a password. For some products this is necessary, for others this necessity is debatable. A citizen who does not have a DigiD yet, who wants to report a broken lamp post, will not request a DigiD for this. The process is too complicated and the citizen gains too little from going through this.

The second flaw is that not many applications already use DigiD. Most people use it only once a year for their tax receipt, for which it is mandatory. Because of this, the password is easily forgotten.

The final flaw we mention here is that DigiD is a centralised access control application. When it malfunctions, none of the products that use DigiD for access control can be used anymore. Some products may not need the high security that DigiD offers. These products can very well remain functional even without DigiD.

To eliminate these flaws, we set out to develop an alternative way to perform access control. The main idea is to develop a system that selects an appropriate access control mechanism based on the security needs of the product that is requested. Users answer easy questions that they know by heart for products that do not need high security, while products that do need high security can only be accessed when the user provides other credentials that can provide this amount of security. Our main research question asks how we can design a dynamic access control system that takes the information from the communication channel and the requirements of the requested service into account when selecting an appropriate authentication mechanism.

Our main contribution is the dynamic access control system, which consists of three parts. The first part calculates the amount of security that is offered by the credentials that have been entered so far. It does this by looking at the probability that a credential can be cracked or discovered. The second part calculates how much confidence we have in the identity of the person who wants access. It takes into account how common the values are that the user has entered to come up with an identity confidence score. The final part takes the results from the first two parts and decides whether to allow or deny access. When additional credentials are available, it also decides which one to ask. The design of the system is elaborated on in chapters 4, 5 and 6.

We can now answer the research questions that we have posed in chapter 1.

1. *How can we obtain a confidence in the user's identity that is as high as needed, even when the authentication system is presented with incomplete data?*

The system is able to do this by identifying the credential that contains the most unique values. By asking this credential to the user, we are able to identify the user sufficiently as quickly as possible. This is explained in more detail in chapter 5.

2. *How can we have the authentication system automatically assess the amount of risk involved when allowing a certain user to access a service, and take decisions based on this assessment?*

The amount of risk involved is modelled by the security level. Each credential can provide a certain amount of security. This amount of security is assessed by calculating the probability that a credential can be guessed by random guesses and estimating the probability that it can be guessed by using additional knowledge. More details on this can be found in chapter 4.

3. *Which authentication methods are available, which amount of security can they ensure and which communication channels can they be used with?*

Many authentication methods are currently available. We have looked into a number of them in chapter 2. We have assessed the amount of security they can assure by applying our method to calculate the security level. This has been done in chapter 4.

4. *How can we let the authentication system know what input information and security constraints a product needs, so it can use this when taking decisions?*

The security constraints for a product need to be given careful consideration. This needs to be done by requiring an appropriate security level and identity confidence for each product. Determining the appropriate security level and identity confidence needs to be done by a security expert. We elaborate on this in chapter 6.

5. *In what ways can the authentication system adapt itself when an authentication method is unavailable?*

The system is able to select an appropriate authentication method from the methods that are available. Once a method becomes unavailable, the system will no longer choose it. As long as the system knows that the method is not available, it will have no problems authenticating a user using other

authentication methods, provided that these can provide sufficient security and identity confidence.

Users have evaluated the concept by means of a prototype. The participants have used the prototype to log into two products, and filled in a questionnaire afterwards. The results of this questionnaire show that the dynamic access control system successfully adapts to the needed amount of security. Participants value the amount of perceived security approximately the same for all used products. They, however, do not think that the system is easier or more pleasant to use than the widespread access control mechanism of username and password. We believe the main causes for this are the fact that users do not know beforehand how long the login process will take, the possibility that the system asks questions for which they do not know the answers by heart, and the fact that the system of username and password is very common and everyone is very used to it.

We conclude that we have made a sound start with this new way of performing access control. The system is perceived as usable both by participants in the evaluation and security experts. We think this looks promising. A number of issues can be tackled to enhance the system even more. We discuss these now.

10.1 Future work

Evaluate quantified concepts We have split the concepts of security level and identity confidence. As we have seen in section 8.1, this may cause problems in some very specific situations. Carefully setting up the system will avoid these problems. It may, however, be worthwhile to find a solution that makes it impossible for these problems to occur at all. Combining security level and identity confidence into a single concept may be a solution; further investigation is needed.

Enhanced user identification Our system currently identifies users based on values that the system knows for a number of credentials. The system may be able to deduce the value for one credential from the value of another credential. From a user's e-mail address we may deduce his first name, as we have shown in section 5.3. Incorporating this into the system may increase the efficiency of user identification.

Additional usability tests Our evaluation has only touched the surface of usability tests for the system. In section 9.3 we have shown a number of security-related usability tests that already exist. More usability evaluations are available from the field of human computer interaction. It is useful to perform these tests as well to gain more insight in the usability of the dynamic access control system.

Common credentials only A factor that has had a negative impact on the usability of our prototype, according to a number of participants in the evaluation, was that the system could ask them a question for which they do not know the answer by heart. It may be possible that people perceive the system as easier and more pleasant to use when they only need to answer questions that they do not perceive as difficult or hard to find the answer to. Of course, the security implications of such an approach need to be taken into consideration.

Long term evaluation Participants in our evaluation only worked with the system very shortly before they had to answer questions about it. Because the system has a

completely new approach to access control, that the participants were unfamiliar with, they may have rated it lower on pleasantness, ease of use and security. A longer pilot project, during which the participants use the system for a number of common tasks for a longer amount of time, is needed to assess the actual usability of the system. This may change as users become more familiar with it.

Usage with high security credentials Our prototype only worked with answers to questions that users had to type into the system directly. It is possible to use the system with more secure credentials as well. It is interesting to see how users value the security of the system when it asks them to provide a fingerprint for a high security product, or redirects them to DigiD when they request a governmental product that deals with private information.

Investigate secure implementation This research only looks into the principles of dynamic access control. Before the system is actually usable in a production environment, many additional security issues need to be tackled. How does the system obtain its information on credentials? How can we secure the connection between the system and the used data sources? How can the information stored in a secure way, also when the values are in use by the system? We need answers to these questions before the dynamic access control system can be put to actual use.



Instructions

166

Inloggen op internetpagina's gebeurt normaal gesproken door gebruik te maken van een gebruikersnaam en een wachtwoord. Dit onderzoek gaat over de mogelijkheden van een nieuw systeem dat hiervoor gebruikt kan worden: het dynamische toegangssysteem. Voor dit onderzoek gaat u gebruik maken van dit nieuwe systeem.

Dit formulier bevat de instructies voor het gebruik van dit systeem en de gegevens die u hiervoor nodig heeft. U gaat namelijk niet als uzelf inloggen, maar als een verzonden gebruiker van het systeem. De gegevens voor deze gebruiker staan onder de instructies.

Wanneer u wilt kunt u dit formulier uitprinten zodat u de gegevens eenvoudig bij de hand heeft tijdens het werken met het prototype.

De deelname aan het onderzoek duurt ongeveer een kwartier.

Instructies

1. Bezoek de internetpagina <http://emho.protolab.excellence.nl/>
2. Vul in het eerste scherm het nummer in dat u linksboven op dit formulier vindt en klik op "Invoeren".
3. Nu begint het inlogproces. Probeer met de onderstaande gegevens in te loggen om gebruik te maken van het product "Meld kapotte straatverlichting". Om te beginnen klikt u op de naam van het product.

Appendix A

4. Wanneer het inloggen is gelukt, ga dan terug naar het startscherm door te klikken op “Terug naar de lijst met producten” en probeer vervolgens in te loggen om gebruik te maken van het product “Afspraak maken”.
5. Wanneer u dit heeft voltooid kunt u de vragenlijst invullen.

Gebruikersgegevens

Voornaam	Jan
Achternaam	Meerwijck
Postcode	9639
Huisnummer	13
Geboortegemeente	Berkensveen
Telefoonnummer	119452
Paspoortnummer	5695233
Burgerservicenummer	10038596
Gebruikersnaam	jmeerwijck
Wachtwoord	hYe3EVE4
Toegangscade	3942

B

Questionnaire

The following pages contain the questionnaire that participants to the evaluation filled in after they worked with the prototype. The actual questionnaire was conducted online. Participants were directed to the questionnaire immediately after they finished working with the prototype.

271

Vragenlijst dynamisch toegangssysteem

Zojuist heeft u ingelogd met het dynamische toegangssysteem. Uw ervaringen met dit systeem zijn belangrijk voor dit onderzoek. Zou u daarom onderstaande vragenlijst in willen vullen? Uw antwoorden worden anoniem verwerkt.

1. Hoe veel computerervaring heeft u?
 - Heel veel ervaring
 - Veel ervaring
 - Niet veel en niet weinig ervaring
 - Weinig ervaring
 - Heel weinig ervaring
 2. Hoe vaak maakt u gebruik van een computer?
 - Meerdere keren per dag
 - Een enkele keer per dag
 - Een enkele keer per week
 - Minder vaak
 3. Wanneer heeft u voor het eerst gebruik gemaakt van een computer?
 - Minder dan vijf jaar geleden
 - Tussen vijf en tien jaar geleden
 - Tussen tien en vijftien jaar geleden
 - Langer dan vijftien jaar geleden
 4. Hoe vaak bezoekt u internetpagina's?
 - Meerdere keren per dag
 - Een enkele keer per dag
 - Een enkele keer per week
 - Minder vaak
 5. Hoe vaak maakt u gebruik van internetpagina's waar u dient in te loggen?
 - Meerdere keren per dag
 - Een enkele keer per dag
 - Een enkele keer per week
 - Minder vaak
 6. Noteert u wel eens een wachtwoord?
 - Ja
 - Nee
-

Questionnaire

7. Het dynamische toegangssysteem stelde u een aantal vragen. Aan de hand van uw antwoorden heeft het bepaald of u toegang kreeg tot het product. Vindt u deze vragen logisch?
- Ja
 - Nee
 - Weet niet
8. Het dynamische toegangssysteem is bedoeld om ervoor te zorgen dat u de enige bent die onder uw identiteit toegang kan krijgen tot producten. Heeft u het gevoel dat de vragen die het systeem stelde voor voldoende beveiliging zorgen voor het eerste product dat u heeft aangevraagd?
- Ja
 - Nee
 - Weet niet
9. Wat vindt u van de hoeveelheid beveiliging van het dynamische toegangssysteem voor het eerste product dat u heeft aangevraagd? Het systeem is. . .
- veel te veilig
 - te veilig
 - precies goed
 - te onveilig
 - veel te onveilig
10. Het dynamische toegangssysteem is bedoeld om ervoor te zorgen dat u de enige bent die onder uw identiteit toegang kan krijgen tot producten. Heeft u het gevoel dat de vragen die het systeem stelde voor voldoende beveiliging zorgen voor het tweede product dat u heeft aangevraagd?
- Ja
 - Nee
 - Weet niet
11. Wat vindt u van de hoeveelheid beveiliging van het dynamische toegangssysteem voor het tweede product dat u heeft aangevraagd? Het systeem is. . .
- veel te veilig
 - te veilig
 - precies goed
 - te onveilig
 - veel te onveilig
12. Het systeem stelt u in staat meerdere producten aan te vragen. Wanneer u eerst een product aanvraagt dat een laag beveiligingsniveau nodig heeft, en daarna een product dat een hoger beveiligingsniveau nodig heeft, stelt het systeem u tussendoor extra vragen. Dit gebeurt om het hogere beveiligingsniveau te bereiken. Vindt u deze extra vragen storend?
- Heel storend
 - Storend
 - Neutraal
 - Niet storend

Appendix B

- Helemaal niet storend
 - Weet niet
13. Bij huidige inlogsystemen weet u van te voren wat er aan u gevraagd gaat worden, bijvoorbeeld een gebruikersnaam en wachtwoord. Bij het dynamische toegangssysteem weet u dit niet van te voren. Vindt u dit vervelend?
- Helemaal niet vervelend
 - Niet vervelend
 - Neutraal
 - Vervelend
 - Heel vervelend
 - Weet niet

In de volgende vragen gaan we het dynamische inlogstelsel vergelijken met een aantal bestaande systemen. Deze vragen gaan over hoe u de hoeveelheid beveiliging ervaart die verschillende inlogsystemen bieden. Onder "hoeveelheid beveiliging" verstaan wij hoe groot u denkt dat de kans is dat iemand anders onder uw identiteit toegang kan krijgen tot het systeem.

De volgende vragen gaan over een systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord.

14. Hoe ervaart u de hoeveelheid beveiliging die dit systeem biedt?
- Heel onveilig
 - Onveilig
 - Niet veilig en niet onveilig
 - Veilig
 - Heel veilig
 - Ik heb het nog nooit gebruikt
15. Heeft u het gevoel dat het dynamische toegangssysteem veiliger is dan een systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord?
- Veel veiliger
 - Veiliger
 - Even veilig
 - Onveiliger
 - Veel onveiliger
 - Weet niet
16. Zou u een aantal producten aangevraagd. Stel dat hierbij, in plaats van het dynamische toegangssysteem, het systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord was gebruikt. Wat vindt u dan van de veiligheid die een dergelijk systeem biedt voor het eerste product dat u heeft aangevraagd?
- Veel te veilig
 - Te veilig
 - Precies goed
 - Te onveilig

- Veel te onveilig
 - Weet niet
17. Wat vindt u van de veiligheid die het systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord biedt voor het tweede product dat u heeft aangevraagd?
- Veel te veilig
 - Te veilig
 - Precies goed
 - Te onveilig
 - Veel te onveilig
 - Weet niet
18. Hoe eenvoudig vindt u het om te werken met een systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord?
- Heel eenvoudig
 - Eenvoudig
 - Niet eenvoudig en niet moeilijk
 - Moeilijk
 - Heel moeilijk
 - Ik heb het nog nooit gebruikt
19. Vindt u het dynamische toegangssysteem eenvoudiger om mee te werken dan een systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord?
- Veel moeilijker
 - Moeilijker
 - Even eenvoudig
 - Eenvoudiger
 - Veel eenvoudiger
 - Weet niet
20. Vindt u het systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord prettig om mee te werken?
- Heel onprettig
 - Onprettig
 - Niet prettig en niet onprettig
 - Prettig
 - Heel prettig
 - Ik heb het nog nooit gebruikt
21. Vindt u het dynamische toegangssysteem prettiger om mee te werken dan het systeem waarbij u inlogt met alleen een zelfgekozen gebruikersnaam en wachtwoord?
- Veel prettiger
 - Prettiger
 - Even prettig
 - Onprettiger
 - Veel onprettiger
 - Weet niet

Appendix B

De volgende vragen gaan over een systeem dat nummerherkenning gebruikt om u automatisch te herkennen wanneer u gebruik maakt van de telefoon. Dit systeem wordt bij sommige helpdesks gebruikt om automatisch uw gegevens te vinden wanneer u belt.

22. Hoe ervaart u de hoeveelheid beveiliging die dit systeem biedt?
- Heel veilig
 - Veilig
 - Niet veilig en niet onveilig
 - Onveilig
 - Heel onveilig
 - Ik heb het nog nooit gebruikt
23. Heeft u het gevoel dat het dynamische toegangssysteem veiliger is dan een systeem dat nummerherkenning gebruikt?
- Veel onveiliger
 - Onveiliger
 - Even veilig
 - Veiliger
 - Veel veiliger
 - Weet niet
24. Zojuist heeft u een aantal producten aangevraagd. Stel dat hierbij, in plaats van het dynamische toegangssysteem, het systeem met nummerherkenning was gebruikt. Wat vindt u dan van de veiligheid die een dergelijk systeem biedt voor het eerste product dat u heeft aangevraagd?
- Veel te veilig
 - Te veilig
 - Precies goed
 - Te onveilig
 - Veel te onveilig
 - Weet niet
25. Wat vindt u van de veiligheid die het systeem met nummerherkenning biedt voor het tweede product dat u heeft aangevraagd?
- Veel te veilig
 - Te veilig
 - Precies goed
 - Te onveilig
 - Veel te onveilig
 - Weet niet
26. Hoe eenvoudig vindt u het om te werken met een systeem waarbij u inlogt door het gebruik van nummerherkenning?
- Heel eenvoudig
 - Eenvoudig
 - Niet eenvoudig en niet moeilijk

- Moeilijk
 - Heel moeilijk
 - Ik heb het nog nooit gebruikt
27. Vindt u het dynamische toegangssysteem eenvoudiger om mee te werken dan een systeem waarbij u inlogt door het gebruik van nummerherkenning?
- Veel moeilijker
 - Moeilijker
 - Even eenvoudig
 - Eenvoudiger
 - Veel eenvoudiger
 - Weet niet
28. Vindt u het systeem waarbij u inlogt met behulp van nummerherkenning prettig om mee te werken?
- Heel prettig
 - Prettig
 - Niet prettig en niet onprettig
 - Onprettig
 - Heel onprettig
 - Ik heb het nog nooit gebruikt
29. Vindt u het dynamische toegangssysteem prettiger om mee te werken dan het systeem waarbij u inlogt door het gebruik van nummerherkenning?
- Veel onprettiger
 - Onprettiger
 - Even prettig
 - Prettiger
 - Veel prettiger
 - Weet niet

De volgende vragen gaan over een systeem waarbij u dient in te loggen met uw gebruikersnaam en wachtwoord van DigiD, inclusief gebruik van de beveiligingscode via SMS. Nadat u uw gebruikersnaam en wachtwoord heeft ingevuld ontvangt u een SMS met een beveiligingscode. Deze code dient u ook in te vullen om toegang te krijgen.

30. Hoe ervaart u de hoeveelheid beveiliging die dit systeem biedt?
- Heel veilig
 - Veilig
 - Niet veilig en niet onveilig
 - Onveilig
 - Heel onveilig
 - Ik heb het nog nooit gebruikt
31. Heeft u het gevoel dat het dynamische toegangssysteem veiliger is dan DigiD met SMS-functie?

Appendix B

- Veel veiliger
 - Veiliger
 - Even veilig
 - Onveiliger
 - Veel onveiliger
 - Weet niet
32. Zou juist heeft u een aantal producten aangevraagd. Stel dat hierbij, in plaats van het dynamische toegangssysteem, DigiD met SMS-functie was gebruikt. Wat vindt u dan van de veiligheid die een dergelijk systeem biedt voor het eerste product dat u heeft aangevraagd?
- Veel te veilig
 - Te veilig
 - Precies goed
 - Te onveilig
 - Veel te onveilig
33. Wat vindt u van de veiligheid die DigiD met SMS-functie biedt voor het tweede product dat u heeft aangevraagd?
- Veel te veilig
 - Te veilig
 - Precies goed
 - Te onveilig
 - Veel te onveilig
34. Hoe eenvoudig vindt u het om met een systeem waarbij u inlogt met DigiD met SMS-functie te werken?
- Heel eenvoudig
 - Eenvoudig
 - Niet eenvoudig en niet moeilijk
 - Moeilijk
 - Heel moeilijk
 - Ik heb het nog nooit gebruikt
35. Vindt u het dynamische toegangssysteem eenvoudiger om mee te werken dan een systeem waarbij u inlogt met DigiD met SMS-functie?
- Veel moeilijker
 - Moeilijker
 - Even eenvoudig
 - Eenvoudiger
 - Veel eenvoudiger
 - Weet niet
36. Vindt u een systeem waarbij u inlogt met DigiD met SMS-functie prettig om mee te werken?
- Heel onprettig
 - Onprettig
 - Niet prettig en niet onprettig
 - Prettig

- Heel prettig
 - Ik heb het nog nooit gebruikt
37. Vindt u het dynamische toegangssysteem prettiger om mee te werken dan een systeem waarbij u inlogt met DigiD met SMS-functie?
- Veel prettiger
 - Prettiger
 - Even prettig
 - Onprettiger
 - Veel onprettiger
 - Weet niet
-

38. Wat is uw geslacht?

- Man
- Vrouw

39. Wat is uw leeftijd?

40. Wat is uw hoogst voltooide opleiding?

- Basisonderwijs
- Voortgezet onderwijs
- Lager beroepsonderwijs
- Middelbaar beroepsonderwijs
- Hoger beroepsonderwijs
- Universitaire bachelor
- Universitaire master
- Anders, namelijk

41. Indien u opmerkingen heeft over de vragen of het onderzoek, kunt u deze hieronder invullen.

.....

.....

.....

.....

.....

42. U kunt hieronder uw e-mailadres achterlaten indien u een overzicht wil van de resultaten nadat het onderzoek is afgerond. Dit is niet verplicht. Uw e-mailadres zal op geen enkele wijze gekoppeld worden aan uw antwoorden.

.....

Dit is het einde van de vragenlijst. Hartelijk dank voor uw medewerking.



Evaluation credentials

We have selected a number of credentials that can be used with the prototype. First, a number of password-based credentials for which the probability of crack can be calculated using equation 4.2. These are shown in table C.1. We have used the maximum number of attempts $k = 3$ to determine the probability of crack for these credentials. The influence of the probability of discovery on the probability of compromise $\alpha = 1$.

Table C.1: Password-based credentials

Credential	$ \Sigma $	n	$P(D_a)$	Security level
Citizen ID (Dutch: burgerservicenummer)	10	7	0.75	0.124
Telephone number	10	10	0.9	0.046
Passport number	10	7	0.75	0.124
Postal code (numbers only) ¹	10	4	0.9	0.046
Username	62	6	0.9	0.046
Random password	62	8	0.1	0.999
User-chosen access code	10	4	0.5	0.301

¹ Dutch postal codes are of the form 1234AB. We chose to ask only the numbers to avoid input mistakes. Some people tend to enter a space between the numbers and letters while others do not.

The credentials in table C.2 were also used in the evaluation, but equation 4.2 was not a feasible way to calculate the probability of crack for these credentials.

Table C.2: Other credentials

Credential	$P(C_a)$	$P(D_a)$	Security level
First name ¹	$\frac{1}{500000}$	0.9	0.046
Last name ²	$\frac{1}{314000}$	0.9	0.046
Municipality of birth ³	$\frac{1}{500}$	0.75	0.124
House number ⁴	$\frac{1}{150}$	0.9	0.046

¹ KNAW/Meertensinstituut, *Nederlandse Voornamenbank*, <http://www.meertens.knaw.nl/nvb/>, retrieved 8 June 2010

² KNAW/Meertensinstituut, *Nederlandse Familienamenbank*, <http://www.meertens.knaw.nl/nfb/>, retrieved 7 June 2010

³ KNAW, A. van der Meer and O. Boonstra, *Repertorium van Nederlandse Gemeenten 1812-2006*, <http://www.knaw.nl/publicaties/pdf/20061061.pdf>, retrieved 7 June 2010

⁴ We assume the average street to have 150 house numbers.

D

Evaluation users

The prototype contains a number of test users. We have placed the evaluation in the fictitious town of Berkensveen, to allow similarity in a number of credentials.

Table D.1: List of users in the prototype

First name	Last name	Munic. of birth	Postal code	House No.	Passport No.	Citizen ID	Username	Password	PIN	Tel. No.
Jan	Meerwijk	Berkenveen	9639	13	5695233	10038596	jmeerwijk	hYe3EVE4	3942	119452
Melanie	Nietingmans	Berkenveen	9639	13	9938152	53019482	melanie	ng2S2pSF	8507	119452
Esmee	Meerwijk	Berkenveen	9639	13	7403214	28490290	esmeerw	fEw3phJz	9521	119452
Lisa	Achterhofs	Neerwijk	9639	15	7590509	61477039	lisa15	HJ8UJLxy	5375	737832
Piet	Roelofsens	Berkenveen	9638	2	5297261	39847601	roelofsensp	qNWAzCaG	2349	239752
Petra	Roelofsens	Berkenveen	9638	4	5203489	78913754	petra	9QebWHzP	9835	239753
Lucas	Van Grooijens	Driehoven	9639	7	3249582	52349871	lucasvgr	yzRAAW7kd	7956	529875

E

Evaluation results

E.1 Mean perceived security per authentication method

1: very insecure, 5: very secure

Authentication method	Mean	Standard deviation	N
Caller ID	3.04	1.06	51
Username and password	3.38	0.87	60
DigiD with SMS	4.44	0.70	59

E.2 Perceived security needed for product A

1: much too insecure, 3: exactly right, 5: much too secure

Authentication method	Mean	Standard deviation	N
Caller ID	2.92	0.97	38
Username and password	3.42	0.78	40
DigiD with SMS	4.49	0.79	39

E.3 Perceived security needed for product B (1st)

1: much too insecure, 3: exactly right, 5: much too secure

Authentication method	Mean	Standard deviation	N
Caller ID	2.83	0.99	18
Username and password	3.00	0.33	19
DigiD with SMS	4.05	0.69	20

E.4 Perceived security needed for product B (2nd)

1: much too insecure, 3: exactly right, 5: much too secure

Authentication method	Mean	Standard deviation	N
Caller ID	2.18	0.73	17
Username and password	2.72	0.58	18
DigiD with SMS	3.50	0.62	18

E.5 Perceived security needed for product C

1: much too insecure, 3: exactly right, 5: much too secure

Authentication method	Mean	Standard deviation	N
Caller ID	1.97	0.93	39
Username and password	2.48	0.67	42
DigiD with SMS	3.36	0.69	42

E.6 Reference authentication methods: difficulty

1: very easy, 5: very difficult

Authentication method	Mean	Standard deviation	N
Caller ID	2.08	1.00	49
Username and password	1.70	0.74	60
DigiD with SMS	2.59	1.07	59

E.7 Reference authentication methods: pleasant

1: very unpleasant, 5: very pleasant

Authentication method	Mean	Standard deviation	N
Caller ID	3.00	1.00	46
Username and password	3.67	0.88	60
DigiD with SMS	3.21	1.01	57

E.8 Relative security per product for DACS

1: much too insecure, 3: exactly right, 5: much too secure

Product	Mean	Standard deviation	N
Product A	2.42	0.84	40
Product B (1st)	2.35	0.59	20
Product B (2nd)	2.78	0.65	18
Product C	2.61	0.63	41

E.9 Relative perceived pleasantness for DACS

1: much more unpleasant, 3: just as pleasant, 5: much more pleasant

Authentication method	Mean	Standard deviation	N
Caller ID	3.28	1.00	47
Username and password	2.67	0.93	60
DigiD with SMS	3.19	0.96	50

E.10 Relative perceived difficulty for DACS

1: much more difficult, 3: just as difficult, 5: much less difficult

Authentication method	Mean	Standard deviation	N
Caller ID	3.60	0.97	53
Username and password	3.34	0.94	59
DigiD with SMS	2.62	0.98	60

List of definitions

1	Access control	7
2	Authentication	7
3	Authorisation	10
4	Credential	20
5	Probability of crack	21
6	Probability of discovery	24
7	Probability of compromise	24
8	Commonness	38
9	Effort	46

List of symbols

α	Variable to adjust the influence of the probability of discovery on the probability of compromise	26
χ	A communication channel	16
δ	An authorisation decision	12
Σ	Set of characters that together comprise an alphabet	23
π_a	Projection of attribute $a \in \mathbf{A}$	15
\mathbf{A}	Set of attributes that a user can have	15
\mathbf{P}	Set of products	15
\mathbf{U}	Set of users	15
\mathbf{X}	Set of communication channels	15
a	An attribute	15
D_a	The domain of attribute $a \in \mathbf{A}$	15
d_a	A value from the domain D_a of attribute $a \in \mathbf{A}$	15
$R(u)$	Representation of user u in the system	15
u	A user	15
\bar{v}	The average commonness for a credential	40

Bibliography

- [1] A. Adams and M. A. Sasse. Users are not the enemy. *Commun. ACM*, 42(12): 40–46, 1999.
- [2] R. J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley Publishing, 2008.
- [3] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semant.*, 5(2):58–71, 2007.
- [4] Y. Bai and V. Varadharajan. A logic for state transformations in authorization policies. In *CSFW '97: Proceedings of the 10th IEEE workshop on Computer Security Foundations*, pages 173–182, Jun 1997.
- [5] L. Ballard, S. Kamara, and M. K. Reiter. The practical subtleties of biometric key generation. In *SS'08: Proceedings of the 17th conference on Security symposium*, pages 61–74, Berkeley, CA, USA, 2008. USENIX Association.
- [6] J. Bentley and C. Mallows. How much assurance does a PIN provide? In *Human Interactive Proofs*, volume 3517 of *Lecture Notes in Computer Science*, pages 111–126. Springer Berlin / Heidelberg, 2005.
- [7] J. C. Bertot, P. T. Jaeger, and C. R. McClure. Citizen-centered e-government services: benefits, costs, and research needs. In *dg.o '08: Proceedings of the 2008 international conference on Digital government research*, pages 137–142. Digital Government Society of North America, 2008.
- [8] B. Bhargava, L. Lilien, A. Rosenthal, M. Winslett, M. Sloman, T. Dillon, E. Chang, F. Hussain, W. Nejdl, D. Olmedilla, and V. Kashyap. The pudding of trust [intelligent systems]. *Intelligent Systems, IEEE*, 19(5):74–88, Sept.-Oct. 2004.
- [9] E. Bina, R. McCool, V. Jones, and M. Winslett. Secure access to data over the internet. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, pages 99–102, sep 1994.
- [10] P. Bonatti and P. Samarati. Regulating service access and information release on the web. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*, pages 134–143, New York, NY, USA, 2000. ACM.
- [11] P. Bonatti, N. Shahmehri, C. Duma, D. Olmedilla, W. Nejdl, M. Baldoni, C. Baroglio, A. Martelli, V. Patti, P. Coraggio, et al. Rule-based policy specification: State of the art and future work. Report I2-D1, REVERSE, August 2004.

Bibliography

- [12] P. Bonatti, C. Duma, D. Olmedilla, and N. Shahmehri. An integration of reputation-based and policy-based trust management. In *Proceedings of the Semantic Web Policy Workshop*, 2005.
- [13] J. Bonneau, M. Just, and G. Matthews. What's in a name? Evaluating statistical attacks on personal knowledge questions. In *Proceedings of the fourteenth International Conference on Financial Cryptography and Data Security (to be released)*, 2010.
- [14] T. Bridis. Hacker impersonated Palin, stole e-mail password, Sept. 2008. URL http://www.usatoday.com/news/politics/2008-09-17-152224562_x.htm.
- [15] P-C. Cheng, P. Rohatgi, C. Keser, P. Karger, G. Wagner, and A. Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *Security and Privacy, 2007. SP '07. IEEE Symposium on*, pages 222–230, May 2007.
- [16] D. Davis, F. Monrose, and M. K. Reiter. On user choice in graphical password schemes. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 11–11, Berkeley, CA, USA, 2004. USENIX Association.
- [17] M. Dell'amico, P. Michiardi, and Y. Roudier. Measuring password strength: An empirical analysis. Technical report, arXiv, Jul 2009. URL <http://arxiv.org/abs/0907.3402>.
- [18] W. Dijkstra and J. Smit. *Onderzoek met vragenlijsten. Een praktische handleiding*. VU Uitgeverij, 1999.
- [19] N. Dimmock. How much is "enough"? risk in trust-based access control. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*, pages 281–282, June 2003.
- [20] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, and K. Moody. Using trust and risk in role-based access control policies. In *SACMAT '04: Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 156–162, New York, NY, USA, 2004. ACM.
- [21] N. Dimmock, J. Bacon, D. Ingram, and K. Moody. Risk models for trust-based access Control(TBAC). In *Trust Management*, volume 3477 of *Lecture Notes in Computer Science*, pages 364–371. Springer Berlin / Heidelberg, 2005.
- [22] P. Dourish, E. Grinter, J. Delgado de la Flor, and M. Joseph. Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal Ubiquitous Comput.*, 8(6):391–401, 2004.
- [23] E-Authentication Initiative, US. E-authentication guidance for federal agencies, 2003. URL <http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf>.
- [24] A. Elmagarmid, P. Ipeirotis, and V. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, Jan. 2007.

- [25] T. Espiner. RSA launches identity verification product for UK, April 2010. URL <http://www.zdnet.co.uk/news/security-management/2010/04/21/rsa-launches-identity-verification-product-for-uk-40088682/>.
- [26] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
- [27] S. Gaw and E. W. Felten. Password management strategies for online accounts. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 44–55, New York, NY, USA, 2006. ACM.
- [28] P. D. Giang, L. X. Hung, S. Lee, Y.-K. Lee, and H. Lee. A flexible trust-based access control mechanism for security and privacy enhancement in ubiquitous systems. *Multimedia and Ubiquitous Engineering, International Conference on*, 0:698–703, 2007.
- [29] T. Grandison and M. Sloman. A survey of trust in internet applications. *Communications Surveys & Tutorials, IEEE*, 3(4):2–16, Quarter 2000.
- [30] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: the teenage years. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 9–16. VLDB Endowment, 2006.
- [31] C. Irvine and T. Levin. Quality of security service. In *NSPW '00: Proceedings of the 2000 workshop on New security paradigms*, pages 91–99, New York, NY, USA, 2000. ACM.
- [32] S. Jajodia, P. Samarati, and V. Subrahmanian. A logical language for expressing authorizations. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 31–42, May 1997.
- [33] D. L. Jobusch and A. E. Oldenhoef. A survey of password mechanisms: weaknesses and potential improvement, part 1. *Comput. Secur.*, 8(7):587–601, 1989.
- [34] A. Jøsang. A subjective metric of authentication. In *Computer Security – ESORICS 98*, volume 1485 of *Lecture Notes in Computer Science*, pages 329–344. Springer Berlin / Heidelberg, 1998.
- [35] P. Kotler, G. Armstrong, L. Brown, and S. Adam. *Marketing*. Pearson Education Australia/Prentice Hall, 7th edition, 2006.
- [36] Logius, Dutch government organization for ICT implementation. DigiD certainty levels. URL <http://www.logius.nl/producten/toegang/digid/productinformatie/zekerheidsniveaus/>.
- [37] R. McKenzie, M. Crompton, and C. Wallis. Use cases for identity management in e-government. *IEEE Security and Privacy*, 6(2):51–57, 2008.
- [38] P. Nagabhushan, S. Angadi, and B. Anami. A soft computing model for mapping incomplete/approximate postal addresses to mail delivery points. *Applied Soft Computing*, 9(2):806 – 816, 2009.

Bibliography

- [39] New Zealand e-Government Programme. Authentication: final analysis of services data by trust levels, 2008. URL <http://www.e.govt.nz/services/authentication/authentication-trust-levels/authentication-trust-levels-report.pdf>.
- [40] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. *Science*, 130(3381):954–959, 1959.
- [41] X. Ni and J. Luo. A trust aware access control in service oriented grid environment. In *Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on*, pages 417–422, Aug. 2007.
- [42] Office of the e-Envoy, UK. Registration and authentication - e-government strategy framework policy and guidelines version 3.0, 2002. URL http://www.cabinetoffice.gov.uk/govtalk/policydocuments/security/security_framework/registration_and_authentication.aspx.
- [43] L. O’Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, November 2004.
- [44] M. Qadeer, M. Salim, and M. Akhtar. Profile management and authentication using LDAP. In *ICCET ’09: Proceedings of the 2009 International Conference on Computer Engineering and Technology*, volume 2, pages 247–251, Washington, DC, USA, Jan. 2009. IEEE Computer Society.
- [45] D. Recordon and D. Reed. OpenID 2.0: a platform for user-centric identity management. In *DIM ’06: Proceedings of the second ACM workshop on Digital identity management*, pages 11–16, New York, NY, USA, 2006. ACM.
- [46] P Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems: facilitating trust in internet interactions. *Communications of the ACM*, 43(12): 45–48, 2000.
- [47] RSA. Identity verification. URL <http://www.rsa.com/node.aspx?id=3347>.
- [48] M. Sahinoglu. Security meter: a practical decision-tree model to quantify risk. *Security Privacy, IEEE*, 3(3):18–24, May-June 2005.
- [49] R. Sandhu and P. Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40–48, Sep 1994.
- [50] SANS Institute. Glossary of terms used in security and intrusion detection. URL <http://www.sans.org/security-resources/glossary.php>.
- [51] S. Schechter, A. Brush, and S. Egelman. It’s no secret. Measuring the security and reliability of authentication via ‘secret’ questions. In *SP ’09: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pages 375–390, May 2009.
- [52] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2004.
- [53] J.-M. Seigneur, S. Farrell, C. D. Jensen, E. Gray, and Y. Chen. End-to-end trust starts with recognition. In *Security in Pervasive Computing*, volume 2802 of *Lecture Notes in Computer Science*, pages 251–255. Springer Berlin / Heidelberg, 2004.

- [54] X. Suo, Y. Zhu, and G. S. Owen. Graphical passwords: a survey. In *ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, pages 463–472, Dec. 2005.
- [55] I. Thomas, M. Menzel, and C. Meinel. Using quantified trust levels to describe authentication requirements in federated identity management. In *SWS '08: Proceedings of the 2008 ACM workshop on Secure web services*, pages 71–80, New York, NY, USA, 2008. ACM.
- [56] M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic XML approach to data integration. In *Proceedings of the 21st International Conference on Data Engineering (ICDE'05), Tokyo, Japan*, IEEE Conference Proceedings, pages 459–470, Washington, DC, USA, April 2005. IEEE Computer Society.
- [57] N. N. Vuong, G. S. Smith, and Y. Deng. Managing security policies in a distributed environment using extensible markup language (XML). In *SAC '01: Proceedings of the 2001 ACM symposium on Applied computing*, pages 405–411, New York, NY, USA, 2001. ACM.
- [58] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Negotiating disclosure of sensitive credentials. In *Second Conference on Security in Communication Networks*, Amalfi, Italy, September 1999.
- [59] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, volume 1, pages 88 –102 vol.1, 2000.
- [60] T. Y. Woo and S. S. Lam. Authorization in distributed systems: A formal approach. In *SP '92: Proceedings of the 1992 IEEE Symposium on Security and Privacy*, page 33, Washington, DC, USA, 1992. IEEE Computer Society.
- [61] H. M. Wood. The use of passwords for controlling access to remote computer systems and services. In *AFIPS '77: Proceedings of the June 13-16, 1977, national computer conference*, pages 27–33, New York, NY, USA, 1977. ACM.
- [62] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, 2:25–31, 2004.
- [63] K.-P. Yee. User interaction design for secure systems. In *ICICS '02: Proceedings of the 4th International Conference on Information and Communications Security*, pages 278–290, London, UK, 2002. Springer-Verlag.
- [64] M. Zurko, R. Simon, and T. Sanfilippo. A user-centered, modular authorization service built on an RBAC foundation. pages 57 –71, 1999.