Radboud University Nijmegen

Faculty of Science

Battling Disinformation: Towards Comprehensible Digital Signature Semantics

An attribute-based digital signature toolset using IRMA

Master Thesis Computing Science

Author:
Martijn J.G. van Dijk

Supervisor:
prof. dr. Bart P.F. Jacobs

Second reader:
dr. Jaap-Henk Hoepman

May 2021

Abstract

In the battle against disinformation many countermeasures have been proposed, one of them being the use of digital signatures. When someone wants to guarantee the authenticity of some content, they can enrich this content with a digital signature. This guarantee of authenticity can be helpful in the battle against disinformation. Instead of claiming the correctness of the content, proving the authenticity of the source gives the reader a better chance to value the content. However, the problem that currently arises when using digital signatures is the lack of comprehensible semantics. The information displayed on a digital signature is often very technical and difficult to understand for the average person. This lack of comprehensibility holds back the adoption of digital signatures in people's day-to-day lives and consequently fails to support the battle against disinformation. A potential solution to increase the comprehensibility of digital signature semantics is by introducing so-called attributes. These attributes provide (personal) information about the author (signer), giving the verifier of the signature a better understanding of the author's identity. This thesis aims to provide both a theoretical and practical solution to the lack of comprehensible digital signature semantics. Through fundamental and applied research on attribute-based technologies and studying existing identity-based systems like IRMA, a toolset is developed that realizes the use of attribute-based signatures in practice. The results will show that signing digital content using attribute-based signatures enables the user to better define the authenticity of a source and thereby provide support in the battle against disinformation. This thesis will conclude what has been achieved throughout the research and what the contribution has been with regard to the field of computing science.

Acknowledgments

I would first like to thank my supervisor *Bart Jacobs* who provided me with valuable knowledge and feedback, using his expertise to steer me in the right direction where necessary. I would also like to thank *Jaap-Henk Hoepman* for introducing me to the subject of this thesis and bringing me in contact with Bart Jacobs. I feel truly honoured to have worked together with both Bart and Jaap-Henk. Not everyone is given the chance to be supervised by such knowledgeable people.

In addition, I would like to thank everyone else who provided support throughout my student time at the Radboud University, in particular those who provide me with the necessary distractions outside my study and researches.

Contents

List of Figures

List of Tables

1 Introduction

Everyone should have the right to freedom of speech. Freedom of speech is a fundamental human right enshrined in Article 19 of the Universal Declaration of Human Rights. Since the introduction of social media, this right has been stronger than ever. This does not mean that the current social media landscape is perfect. Problems like social media platforms using algorithms to prioritize posts are still an issue. However, publishing your thoughts to millions of people has never been easier. This ease of publishing does, on the other hand, come at a cost. The amount of 'false' and 'misleading' information that reaches the masses has also increased.

Since the introduction of the internet, particularly social media, the ease by which someone can spread disinformation has sky-rocketed. It used to be the established newspapers that would mainly 'inform' the general public; everyone can now raise their voice using platforms like Twitter or Facebook. While this provides social benefits related to freedom of speech and democratic participation, the disadvantages seem to become more evident. Being able to 'shout' everything you want to a massive (digital) public can have consequences. Like those with many (digital) followers, influential people can spread any information they want, perhaps without realizing the potential consequences.

Countermeasures have been proposed to battle against the impact or spread of disinformation. However, the battle is often fought from a 'correctness' or 'truth' perspective, where the information is deemed as 'false' or 'fake'. Relating to the introduction of the famous term *fake-news*. These correctness-based countermeasures seem to focus on changing the information, or otherwise condemning the information as 'fake', while not providing the reader the ability to validate the source. In other words, the discussion when trying to fight disinformation should shift from the correctness of information to the authenticity of the source. When the reader of (dis)information can hold the author responsible for what is written, the reader can make his own decision whether or not to trust the received information. When exclusively claiming that certain information is false or fake, it is more difficult for the reader to form his own thought. Instead you can provide the authenticity of the source, thereby you give the reader a fair chance to form his own thought without altering the content.

One of the most promising countermeasures that focus on authenticity rather than correctness is the *digital signature*. Digital signatures allow information to be signed by the author. The reader can consequently validate which information originates from which author. Imagine an author writing a column with his personal opinions weaved into the text. When the column is spread by the author (writer), the reader of this column can validate that the text is indeed written by this specific author. Whether or not the information is correct can be decided by the reader. Moreover, the reader can be sure of the identity of the author and hold the author responsible for what has been written. On the other side we have the author, for the author it is conforming to know that precisely what was written is picked up by the reader, not a malicious version of the column altered along transition.

However, one major problem with the current practical implementations of digital signatures is the lack of comprehensibility for the ‘average’ user. It is not easy to understand most of the digital signature semantics when you are not a tech-savvy person. Especially when the signature semantics includes technical information like the hash, the signer’s public key, and the name of a certificate authority. For digital signatures to integrate into the day-to-day life of the ‘average’ internet user, the semantics of the signature should be easy to understand. As a solution to this lack of comprehensibility, we will look into the use of *attribute-based signatures* as a technique that can provide authenticity to digital content, make the semantics of a digital signature more comprehensible, and provide a higher level of information value on a digital signature. All while aiming to preserve the user’s privacy, since disclosing more about yourself could lead to less privacy. We see how this is solved by letting the user selectively disclose (personal) attributes.

This thesis explores the possibility of a potential theoretical solution to comprehensible digital signature semantics and tries to translate the solution to practical steps. Starting with theoretical research, followed by applied research. We will look into using the identity platform named IRMA, developed by the Privacy by Design Foundation, to reach comprehensible digital signature semantics. IRMA will function as a basis to develop a new toolset, enabling the signage of digital content using attribute-based signatures. The thesis will end with a few use-cases that describe how the toolset, and attribute-based signatures in general, can be used to support the battle against disinformation, and what the future potential of IRMA, the toolset, and attribute-based signatures in general can be.

1.1 Research Questions

This master thesis will investigate what kind of toolset should be developed to realize comprehensible digital signatures, thereby providing help in battling the negative impact of disinformation. The main research question has been defined as follows:

> MQ: *What kind of toolset should be developed to increase the comprehensibility of digital signatures semantics, in particular to support the battle against disinformation?*

In order to properly answer this main research question, the following sub-questions will be answered:

- Q1: *How can attribute-based signatures be used to achieve comprehensible digital signature semantics?*
- Q2: *How to increase the informational value of digital signatures?*
- Q3: *What role can IRMA play in realizing the toolset?*
- Q4: *What are the technical requirements and related security properties when realizing the toolset?*
- Q5: *What kind of application(s) should be designed and developed as part of the toolset to realize accessible and comprehensible digital signatures?*

2 Methods

This thesis consists of two parts, (1) fundamental research and (2) applied research. Fundamental research is performed to understand the past and current developments related to disinformation and the theoretical possibilities of (attribute-based) digital signatures. Applied research is used to translate the fundamental research to a usable toolset, supporting the fundamental research and vice versa. The two research methods will intertwine throughout the *Research*, *Report*, and *Develop* phases. The research starts off mostly in fundamental research, followed by a combination of both applied and fundamental research.

Throughout the research, more details on how to develop the (ideal) toolset will become clear, which leads to a set of requirements for the toolset to satisfy. The MoSCoW requirements prioritization method is used as part of the applied research to keep the scope reasonable regarding the development of the toolset. And to decide which functionalities of the toolset should (at minimum) be implemented to satisfy the requirements before the toolset can be considered a *minimal viable product* (MVP) and (publicly) deployed.

2.1 Strategy

This thesis will start with a reconnaissance of past developments within the field of disinformation. Gathering knowledge about what impact the spread of disinformation has and could have on society, what countermeasures have been proposed in the past, and which of these proposed countermeasures seems most reasonable. Collecting enough information about theoretical solutions to the problem of disinformation, followed by defining a potential practical solution. The goal of the practical solution is to show how the theoretical solution can be realized in practice. By not only showing the theoretical solution but also including a practical implementation of the solution, it gives this thesis more value regarding the idea of what could be possible in a practical sense.

2.1.1 Collection of Information

The fundamental research collects information on the following topics:

- **Disinformation**: the history of disinformation and its current impact on society.
- **Countermeasures**: proposed countermeasures to battle the spread as well as the impact of disinformation, including preliminary research.
- **Digital signatures**: details about the use of (attribute-based) digital signatures to battle the impact of disinformation.
- **Informational value**: information on how to increase the informational value of digital signatures semantics.
- **IRMA**: details about the theoretical solutions IRMA can deliver. Including details about a potential practical realization of the attribute-based signature toolset that uses IRMA as its basis.

The applied research collects the following information:

- **Toolset**: information on how to design and develop a practical solution (toolset) that increases the level of comprehensibility of digital signature semantics.
- **Requirements**: minimal requirements of the developed applications (toolset), and potential future requirements that improve on the (in this thesis) developed toolset.
- **Technical Design Principles**: choices made to make the toolset easy-in-use, make the semantic of digital signatures comprehensible, and the toolset accessible in general.
- **Future work & research**: potential future work and research on the toolset, and the increased accessibility and comprehensibility of (attribute-based) digital signatures in general.

2.1.2 Prioritization of Requirements

In order to keep the (practical) scope of this thesis reasonable and clear, the MoSCoW requirements prioritization method is used. The method facilitates the ability to scope the development of the toolset, define what the minimal viable product (MVP) is (before it satisfies the set goal of this thesis), and what future work there could be to improve the final product. The MoSCoW method is split up between four different requirements 'levels', from most important (MUST) to least important (WOULD):

- **MUST HAVES**: requirements that must be present in the final product. Without them it would defeat the purpose of the related project.
- **SHOULD HAVES**: requirements that should be present in the final product, but the product can be deployed without them. Non-essential but preferred to be present.
- **COULD HAVES**: requirements that could be present in the final product. Can be deployed without them, and are non-essential to the functioning of the final product. The realization of these requirements are seen as a 'bonus'.
- **WOULD HAVES**: requirements that would be nice to have in future versions of the product. Likely unrealistic to be realized within the current project.

2.2 Structure

The thesis is split up in several chapters, The following enumeration describes the content of each chapter, excluding *Abstract*, *Acknowledgements*, *Definitions*, *Introduction*, *Methods*, and *Conclusion*.

1. Disinformation & Modern Times: this chapter describes the rising spread and impact of disinformation in modern and past times. The focus is on the modern countermeasures, why some of these countermeasures are not sufficient in the battle against disinformation, and what can be the alternative solution.

2. Comprehensibility of Digital Signatures: this chapter describes how the comprehensibility of digital signature semantics can be increased and what preliminary research has been performed related to attribute-based technologies. A link will be provided between the increased comprehensibility and the preferred higher level of informational value of digital signature semantics. The eIDAS regulation will also be discussed to define what the relevant legal obligations are regarding digital signatures.

3. Introduction to IRMA: this chapter describes the background of the IRMA project and what this project is trying to achieve. The current developments within IRMA are discussed, how the project guarantees a high level of privacy friendliness, and what the role of IRMA can be to achieve the goal of this thesis.

4. Technical Specifications of IRMA: this chapter describes all relevant technical details related to the IRMA project. This mainly concerns the techniques behind the realization of attribute-based (privacy-enhancing) technologies.

5. Security Properties and Guarantees: this chapter describes the security properties and guarantees that IRMA provides, relating to both authentication and signage. Furthermore, the current limitations of using IRMA as a basis for the attribute-based signature toolset are described.

6. Toolset Implementations: this chapter describes how the toolset is realized. This includes the functionalities, the set requirements, and the final results of the IRMA Signature Application. Including the design of a proof-of-concept IRMA Signature Plugin. Finally, the results are combined, which produces a minimal viable product of the toolset.

7. Discussion: this chapter describes multiple potential use-cases for the developed toolset and attribute-based signatures in general. The use-cases give an idea of how the toolset can be used in practice. Furthermore, this chapter describes what further research can be performed to extend this thesis's final product. Alternative projects that try to reach similar goals are also discussed to see the potential of using attribute-based signatures.

2.3 Scope

The scope of this *thesis* includes the search for potential solutions that can help in the battle against the negative impacts of disinformation. This is aimed to be achieved by looking at the theoretical and practical possibilities of using attribute-based technologies. More specifically, we want to find a practical implementation of attribute-based signatures that help users to comprehend the semantics of a digital signature. Ultimately, the increased comprehensibility should lead to the adoption of (attribute-based) digital signatures into the daily life of 'average' internet users.

The focus of the *research* will be on the technical possibilities of using attribute-based signature technology provided by IRMA. Like the development of an attribute-based signature toolset. The requirements of the toolset should be reasonable to realize within the given time-frame of a master thesis, which translates into the scope of the research excluding the focus on 'soft' possibilities like user experience. Research on whether or not the toolset (final product) conforms to all legal obligations and if the toolset would actually be used in-practice is also considered out of scope. We will however discuss what the possibilities are when using the toolset and what it could potentially deliver to the society when adopted (as mainstream).

3 Disinformation and Modern Times

We have all heard or read about some form of *disinformation*, whether it be fake-news, deep-fakes or any other given information that turned out to be a premeditated lie. The introduction of the world wide web, combined with social media, has made the spread of disinformation an ultimate method to achieve several unethical goals. These goals can consist of financial gains like influencing stock markets, criminal activities like fake websites and internet frauds, or influencing public opinion by spreading fake-news or deep-fakes.

It is important to notice that there is a difference between the two terms *disinformation* and *misinformation*. The term disinformation, as stated in 'What is Disinformation?' by Fallis [Fal15], is defined as "misleading information that has the function of misleading". When disinformation is spread, the function of the information is to mislead people deliberately. The term misinformation, on the other hand, stands for spreading misleading information while the author has no intention to do so. Within this thesis, potential solutions are discussed to reduce the spread and impact of both disinformation and misinformation. Ideally, it should not matter whether the information is misleading on purpose or not; both should be battled against equally.

3.1 Modern Challenges

Disinformation is nothing new, throughout history there have been several occasions where the spread of disinformation made a big impact on society. One of the most notable and influential examples is Operation Bodyguard. In World War II a disinformation campaign by the name Operation Bodyguard intended to conceal the planned location of a potential beachhead by the Allies, famously known as D-Day. The allies composed deceiving military reports and sent out fake radio transmissions. The goal was to convince the Germans that a beachhead was being planned at Calais, instead of Normandy. The spread of disinformation is not an unknown technique to the military and intelligence agencies, and as the Operation Bodyguard example portrays, it is a technique used for some time now. Modern examples are often on a somewhat smaller scale, like false stories on news websites or the manipulation of images. Though, these small-scale examples could ultimately lead to a big chain reaction of destabilizing events.

3.1.1 Fake-News

Disinformation comes in different forms, one of these is *fake-news*. The amount of fake-news has seen a strong increase since the introduction of social media. Allowing everyone to spread (dis)information to a big audience with minimal effort, while the source of information can remain hidden. One of the biggest impacts fake-news has is related to the political playing field, influencing public opinion to gain popular support on certain issues or party popularity in general. The spread of fake-news is, as we have seen many times before e.g. regarding the European Union[1], not bound to national borders. The impact of fake-news goes far beyond any national border and influences foreign politics all over the world

[1] Action plan disinformation commission contribution European council (2018)

Fake-news has the potential to spread unrest among the population and even make established news outlets to be distrusted. During the presidency of Donald J. Trump (2017 - 2021) the term fake-news has been brought to light many times. Even a study has been performed by Gunther et. al. [GBN18] where it states that fake news may have contributed to Trump's 2016 presidential election victory. This, and many more controversies, gives a potential rise to the number of Americans, and most likely people outside the United States, losing trust in the established news outlets and perhaps even questioning the value of democracy. One of the consequences of this decline in trust is that people start searching for alternative channels to read their daily news. Social media combined with 'smart' algorithms have allowed us to do so, potentially leading us to a (filter) bubble where your beliefs are always confirmed. Creating parallel societies where people live in a (totally) different reality.

There are many more researches and initiatives on (reducing) the spread and impact of fake-news like the European Union proposing a "EU-wide Code of Practice"[2], Facebook's initiative to assess social media's impact on elections[3] and 'stop' fake-news in general[4]. Other initiatives like MisinfoWeb[5] and EUvsDisInfo[6] try to aggregate research and (media) articles to map the developments regarding disinformation.

3.1.2 Deep-Fakes

One of the latest developments within the sphere of disinformation is the introduction of *deep-fakes*. Deep-fake, a portmanteau of deep-learning and fake, is an automated image or video manipulation technique called synthetic media. The technique, that makes use of machine learning and artificial intelligence, allows identifiable parts of someone's body to be replaced by body features from someone else. E.g. replacing the face of someone in a video with the face of a celebrity and manipulating the voice in the original video to make it seem like it's the celebrity's voice. As you can imagine, this can lead to very disturbing situations where the face and voice of someone, who is probably not acquainted with this deep-fake, is used to make it seem that the information given in the image or video is communicated by the 'deep-faked' person. Therefore, deep-faking is a very useful technique to spread disinformation and to shape disinformation as trustful as possible. Deep-fakes can, among other things, morph the face and manipulate the voice of someone popular within a video, making the video more likely to be believed and trusted, while being disinformation. Future developments related to deep-fakes are not looking so bright either. It becomes easier by the day to create a deep-fake image or video. While in the past it used to be a technique that required some image and video forgery skills, the technique is now widely available to the public[7] [PGC$^+$20]. A minor level of knowledge regarding image and video forgery skills is required at current and future times.

[2] https://ec.europa.eu/commission/presscorner/detail/en/IP$_1$8$_3$370

[3] https://about.fb.com/news/2018/04/new-elections-initiative/

[4] https://www.facebook.com/formedia/blog/working-to-stop-misinformation-and-false-news

[5] https://www2018.thewebconf.org/program/misinfoweb/

[6] https://euvsdisinfo.eu/

[7] Apps like DeepFaceLab or FaceSwap

3.2 Countermeasures

Several countermeasures have been introduced to battle the impact of disinformation. Countermeasures like fact-checking [GC16, GG12] and counter-propaganda [Cul15]. Many countermeasures try to define whether some information is factual or not. In the case of fact-checking, this notion of establishing the 'truth' is strongly present. Parties who claim to use counter-propaganda try to battle circulating propaganda by flooding the media with their 'truth'. We will see why this approach of establishing the truth is not ideal and perhaps undesirable. As an alternative, we will show why giving people a tool to decide for themselves whether or not the information is to be trusted makes for a more reasonable approach to the battle against the negative impact of disinformation.

3.2.1 Fact-Checking

One of the most common countermeasures against disinformation is fact-checking. In short, the term fact-checking can be defined as the following: verifying the factual accuracy of a certain (informational) message. Fact-checking is often used to assess the 'truth' or 'falsehood' of (political) statements. However, the truth is not always as black-and-white as it is often presented by fact-checkers, especially in the political world where many subjects can be highly complex. Sometimes political statements are not a fact but merely an opinion. It can even be argued that having these differences in opinion is essential for a democracy to function as intended. Not every statement or claim can be checked and answered with a true or false. Sometimes there is more than a binary answer.

'The Epistemology of fact checking' by Uscinski [UB13] states that fact-checking has become an important part of news coverage, mainly within the political side of journalism. The critic which Uscinski gives to this development is that using a technique like fact-checking "employs a variety of objectionable methodological practices". Where a certain statement containing multiple facts is treated as if it can merely contain a single fact. This includes categorizing future predictions on accuracy. Leaving out any possible difference in interpretation of certain facts, making facts unambiguous. The research by Uscinski exposes the shortcomings of trying to 'check' a fact. Making it an insufficient and probably undesirable way of battling disinformation. Apart from the doubtful functioning of checking facts, it neither provides the reader any possibility to form their own thought. Laying the focus purely on trying to guarantee the correctness of information by relying on third parties, the fact-checkers, to check the validity of the content. It will be more beneficial to the reader if the source of the content is known. Shifting the importance of battling disinformation from the notion of content correctness to the notion of source authenticity. The reader is helped in defining his own thought when certain about who wrote the article, without stating whether the content is true or false.

3.2.2 Counter-Propaganda

A bombardment of truth, sometimes put forth as *counter-propaganda*. This term was introduced to describe the intensive attempt to counter disinformation or propaganda. Since propaganda is a pretty heavy term and often brings us back to the European theater of the '30s and '40s of the past century, I prefer the term *truth-bombardment* since counter-propaganda is trying to bombard the general public with the ‘truth’. Since the introduction of social media, the spread of fake-news and disinformation, in general, has seen a steady increase. In the old times, before the introduction of social media, it was much harder for people to raise their voices. Social media has allowed everyone to make a (loud) statement from behind their screen, often in a lazy chair. This has allowed for an immense increase in the spread of fake-news.

An opted countermeasure to battle the great amount of fake-news is by doing exactly the same but with the ‘genuine’-news. Spreading a great amount of ‘genuine’-news, and hope that this news gets picked up by the general public instead of the fake-news. However, this technique of truth-bombardment can also be used by the opposing party. A recent example would be the tragic event of MH17 where a plane was shot down when flying over Ukraine[8]. From the point of view of the ‘western’ media outlets, nations like The Netherlands and the United States, this news was brought as the plane being (assumedly) shot down by Ukrainian separatists, and (assumedly) backed by the Russian government, see the official investigation by the Dutch Safety Board for more details[9]. The other party, in this case the Russian government, claimed this to be false and denied any involvement[10]. Which side is right is not of importance, what is of importance is that both parties do not want to receive a bad name from this event. The ‘west’ tries to report the news to their population from their point of view, while Russia does the same. Though, the opposing governments can also decide to mangle in the news coverage of each other. A common technique when mangling in the news coverage and specifically relevant to this example is the creation of so called *noise*.

When one party is trying to cover-up certain news, a strategy can be to send out (bombard) a lot of other (slightly) altered news articles related to the subject. This creates so called noise. Aiming to make the reader lose their overview on which (news) article can be trusted. If another party tries to counter this by spreading even more news articles, it would only increase the noise. Potentially resulting in readers losing their will to find out what article is trustworthy. As a consequence tragic events like MH17 are more likely to be neglected. This could exactly be the goal of an opposing party who wants to cover up their actions. A bombardment of ‘truth’ is likely not going to help the readers but has the potential to help malicious parties in increasing noise.

In addition to this problem, the reader is unable to be certain if he is reading the original articles and who writes the articles. Both the integrity and authenticity of the news articles are unclear. Here is where the *digital signature* comes in. It is difficult, and perhaps even impossible, to guarantee the ‘truthfulness’ of a news article or any other digital content. We can however guarantee the integrity of the content and authenticity of the source by attaching a digital signature.

3.2.3 Digital Signatures

A potential solution to the impact of disinformation which is often overlooked is the use of digital signatures.[11] The general use of (handwritten) signatures is often related to guaranteeing non-repudiation, integrity, and authenticity of many sorts of agreements, often in the form of a contract. However, the correctness of the content cannot be

[8] https://www.onderzoeksraad.nl/en/page/3546/crash-mh17-17-july-2014

[9] https://www.onderzoeksraad.nl/en/media/attachment/2018/7/10/debcd724fe7breport$_mh17_crash.pdf$

[10] https://apnews.com/article/ukraine-netherlands-kuala-lumpur-malaysia-europe-982d965de9ad7fdf41c47bc5206dc780

[11] https://ibestuur.nl/weblog/teken-tegen-nepnieuws

guaranteed using (handwritten) signatures.

Handwritten signatures have been around for quite some time, allowing writers to confirm their authorship or giving parties the ability to agree on a given contract. The main goal for using these 'old-fashion' written signatures is, until this day, giving a visual and legally bound agreement to whatever there needs to be an agreement on. This way of settling an agreement has become embedded in modern society. When the digital world was, and still is, on the rise it was a logical thought to translate this physical practice to the digital world. The first concept of digital signatures was introduced by Diffie and Hellman [DH76] as the *digital signature scheme*, however the paper only theorized that such scheme existed. They proposed that each user publishes a 'public key' used for validating the signature while keeping a 'secret key' used for producing the signature. In the digital signature scheme the user's signature, for a certain message, is a value that depends on both the message and on the user's secret key, such that anyone can validate the user's signature using the user's public key. While it is easy to validate a signature using the user's public key, it is difficult to forge the user's signature since the secret key is, as the name suggests, 'secret' and necessary to 'forge' the signature. The RSA algorithm introduced by Rivest et. al. [RSA78] allowed for a practical but primitive kind of digital signature.

While handwritten signatures have been around for quite some time and are pretty straightforward in use, the digital variant comes with some underlying technical challenges. Digital signatures require the signer ('owner' of the signature and the one who signs the content) to have their own private and public key pair. These two keys are mathematical inverses of each other, making them uniquely linked. This allows an operation (like signing a document) performed using the private key to be reversed using the public key. In simpler terms, the private key is used for signing, while the public key is used for verifying the digital signature.

The security of the signing process relies on ensuring that the private key is accessible only to the signer, and no one else. Once this security property is guaranteed, the receiver can be sure that only this specific signer, with his or her private key, is capable of singing the document or message. Concurrently, ensuring the security properties of *non-repudiation* and *authenticity* since the signer is the unique owner of the private key and can therefore not deny the action (non-repudiation) and can be linked to the content (authenticity). Assuming the public-key algorithm is secure.

Another property the use of digital signatures guarantees is *integrity.* Note that in this case, integrity relates to keeping the content equal to the original, not whether the content is trustworthy or truthful. It should be computationally infeasible for two different documents or any file format, e.g. text, images, etc., to produce the same digitally 'enciphered' content when signed. Most modern signature schemes ensure that even a 1-binary bit of change within the content produces a completely different digital signature. Every change made to the content of the document is recognized and makes the original digital signature invalid, requiring the data to be signed again. The notion of integrity, non-repudiation, and authenticity combined makes the use of digital signature a good contender in the battle against disinformation.

4 Comprehensibility of Digital Signatures

Many digital signature projects can be found on the World Wide Web (WWW), most of them focus on supporting companies in the mission to reduce paperwork. Replacing the need to print a document, put a handwritten signature on the document, and scanning it back to a digital format. This tedious (physical) process is not desired and is therefore often replaced by a digital signature variant.

Within this thesis we like to stick to publicly available projects. In other words, projects who publish software in open-source format. We see this doctrine in projects like the Digital Signature Serviece (DDS) or eSignature[12] by CEF Digital, supported by the European Union. But also the OASIS Digital Signature Services (DSS)[13] by OASIS Open, and IRMA which is developed by a non-profit foundation named Privacy By Design[14]. For reasons discussed in chapter 5 'Introduction to IRMA' the decision is made to use IRMA as a basis to develop a new toolset (that enables users to sign digital content using attribute-based signatures).

There is one (social) aspect that many current implementations of digital signatures lack, *comprehensibility*. The person verifying signed content needs a sufficient amount of information to determine if the received content is valid. It should be clear who signed the content. This allows the verifier, in our case the reader, to decide for himself what to do with the received content. It is crucial that the reader knows what the source of the content is. Upon this property of authenticity, it should also be clear to the reader what the digital signature and its semantics actually mean. The semantics of the signature should be comprehensible to everyone. If (personal) details about the signer, the related content, and the attached signature can be easily read and understood, the informational value of a digital signature increases, which in turn helps with battling against the impact of disinformation.

4.1 Preliminary Research

In order to find projects that can deliver an increased or equal level of informational value of disclosed information while preserving privacy, we look at relevant research that already tried to achieve this. Including research that is not directly linked to the notion of digital signatures. Data minimization is an important aspect when it comes to increased privacy for users. It is also required by EU law. The minimization of data can however lead to a lower level of informational value related to this data. When less data is disclosed, there could consequently be less informational data available for the receiving end (reader). A balance should be considered between keeping the informational value of the revealed data high, while only disclosing minimal (personal) data. In the case of digital signatures, we do not want to disclose too much (personal) information of the signer, but we do want the disclosed (personal) information to be informative and comprehensible. This is comparable to someone wanting to authenticate oneself to a service provider (e.g. Instagram), without revealing more personal information than strictly necessary. This is where attribute-based credentials can be of relevance.

[12] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eSignature

[13] https://www.oasis-open.org/committees/dss/

[14] https://privacybydesign.foundation/en/

4.1.1 Attribute-Based Credentials

Related to authentication, attribute-based credentials can be used to solve the problem of disclosing too much personal information when authenticating to a service provider, while still keeping the informational value of the disclosed information high. In traditional identity management systems, a trusted identity provider (or issuer) issues authentic attributes to a user, like a driver's license or student card, or any other personal data (attribute) that can be used to prove the user's identity. The identity provider has the responsibility to manage the user's personal data that is required for the identification of the user. The user first needs to authenticate himself to the identity provider, followed by the identity provider sending the user's identity information to the related service provider. An example would be a user logging into the Belastingdienst[15] (tax authorities) using his DigiD[16] account. DigiD takes the role of the identity provider that delivers identity information to the user, and the Belastingdienst as the service provider who requests certain attributes from the user that were delivered by DigiD before. The requested attributes are used to authenticate the user.

Fully identifying yourself is often unnecessary as described by Alpár and Jacobs [AJ13]. It is often the case that only a few personal details (attributes) of the user are required by the the service provider to offer a certain service. An existing, but primarily theoretical, project named *ABC4Trust* aimed to develop an ABC framework based on existing ABC technology [Bra00][CL01]. ABC4Trust is a project funded by the European Commission and worked out in detail by Bichsel et. al. [BCD$^+$14]. The paper by Bichsel states that the goal of ABC4Trust is "to address the federation and interchangeability of technologies that support trustworthy yet Privacy-preserving Attribute-based Credentials (Privacy-ABCs)". The project introduces a proof-of-concept architectural framework that enhances the so-called Privacy Preserving Attribute Based Credentials (Privacy-ABCs) features. Note that the term privacy-ABCs seems to be no different from the term Attribute-Based Credentials (ABCs). Therefore, we will continue to use the term ABC when referring to Attribute-Based Credentials.

A similar but more practical project is IRMA. IRMA, an acronym for *I Reveal My Attributes*, aims to realize the functional potential of Attribute-Based Credentials. For the implementation of Attribute-Based Credentials, IRMA (partially) relies on the Idemix (short for Identity Mixer) identity system developed by IBM Research [CL01][IBM12]. IBM's Idemix attribute-based credential system provides different functionalities for proving the possession of attribute-based credentials and their properties. More details on credentials will be discussed in chapter 6.2 'Attributes & Credentials'

In comparison to ABC4Trust, IRMA is being used in a production environment and not merely theoretical. Another advantage that IRMA has over ABC4Trust is the ability to sign content using Attribute-Based Signatures (ABS). Upon the already existing ABC infrastructure within IRMA, a functional implementation of the Attribute-Based Signature is introduced. In chapter 5 'Introduction to IRMA' we will continue exploring the possibilities of IRMA.

[15] https://www.belastingdienst.nl/

[16] https://www.digid.nl/en

4.2 eIDAS Regulation

In this (sub)section, the current eIDAS regulation is discussed. Later on, in chapter 5.4 'Legal Status of IRMA Signatures', the legal status of an IRMA attribute-based signature is categorized to which security level the signature holds in relation to the eIDAS regulation. By first discussing the view on the legal status of digital signatures by the eIDAS, we can better define the legal status of an IRMA signature later on.

The eIDAS (**E**lectronic **ID**endtification **A**uthentication and trust **S**ervices) [Cou14] is an EU regulation on electronic identification and trust services for electronic transactions within the European Single Market. The regulation includes guidelines on the development and usage of electronic signatures (or e-signatures) and defines both requirements for electronic authentication and electronic signatures. EU member states are enforced to aim for a technology-neutral approach, meaning that as long as a certain technology complies with the regulation, other EU member states should allow it to be used within their services. The technical requirements from eIDAS's implementing regulation which are relevant for electronic signatures, are defined in three levels. These three levels make a distinction between the legal strength of each electronic signature.

4.2.1 Levels of electronic signatures

The eIDAS Regulation defines three levels of electronic signature:

- 'Simple' electronic signature;
- Advanced electronic signatures (AdES);
- Qualified electronic signatures (QES).

Electronic signatures are defined in the eIDAS Regulation as "data in electronic form which is attached to or logically associated with other data in electronic form and which is used by the signatory to sign". The 'simple' level of electronic signature is often seen in e-mails or documents in the form of handwritten or 'wet' signatures, even though simply writing your name under an e-mail could also constitute as an electronic signature. Handwritten signatures are scanned and appended to the related e-mail or document. This form of 'simple' electronic signature is regarded as the lowest level of electronic signature as defined in the eIDAS regulation. This means that it has no legal strength when trying to prove the identity of the author (authenticity of the source) since it cannot, with certainty, be traced back to the original author.

Advanced electronic signatures need to adhere to additional requirements. If the electronic signature wants to qualify as 'advanced' it must be (1) uniquely linked to and capable of identifying the signatory. (2) The signatory is allowed to retain control. (3) Any subsequent change of data should be detectable.

Qualified electronic signatures must adhere to even more requirements upon the previous listed requirements related to the advanced electronic signatures. These additional requirements are: (1) the signature is created by a qualified signature creation device (QSCD), like a USB token. (2) The signature is based on a qualified certificate (for electronic signatures). Furthermore, qualified electronic signatures have the same legal standing as handwritten signatures.

4.2.2 Electronic Signatures vs. Digital Signatures

Both qualified electronic signatures and (certificate-based) advanced electronic signatures are considered types of *digital signatures*. Digital signatures, in contrast to 'simple' electronic signatures, use a PKI-based digital certificate to bind an identity to a cryptographic key pair. When a message or document is digitally signed using the private key of the signatory, the authenticity, integrity, and non-repudiation are ensured. When a 'simple' electronic signature is used these three principles can not be ensured.

4.3 Attribute-Based Signatures

Q1: *"How can attribute-based signatures be used to achieve comprehensible digital signature semantics?"*

Traditional digital signatures seem to offer, to a certain extent, the same signing functionality as an attribute-based signature when it dedicates one signing key pair for each role the user wishes to sign under. Like a doctor letting his public key be signed by a medical certification authority, by which the doctor can use his private key to sign content *as a doctor*. If this same person (the doctor) wants to sign some content that does not relate to his medical expertise, there needs to be another key pair under another role e.g. the role of a citizen of a specific nation to buy a house. The role of being a doctor, and thereby the related key pair, is worthless in this situation. You could be a doctor in Germany, but this specific attribute would not give you the right to buy a house in The Netherlands. So, there is a possibility to use traditional digital signatures to 'role-based' sign content, but the process is tedious, difficult to scale and introduces complicated key management issues. In contrast, an attribute-based digital signature enables users to digitally sign content under different roles with a single key pair. The notion of attribute-based signatures have been *explicitly* introduced by Shanqing and Yingpei [SY08]. Maji et. al. [MPR11] continued on this work and describes the attribute-based signature as "a versatile primitive that allows a party to sign a message with fine-grained control over identifying information".

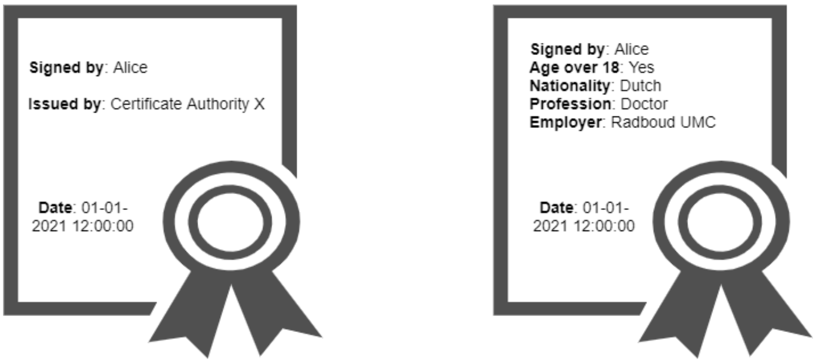
To solve the lack of comprehensible digital signatures we look at the attribute-based signature developed by the Privacy by Design foundation as part of the IRMA project. IRMA uses the attribute-based credential system and expands the identity platform with attribute-based signatures. Translating the functionalities of attribute-based credentials within the process of authentication to the process of signage. The core functionality of IRMA was focused on authentication using attribute-based credentials, but the possibility to translate this functionality to digital signatures opened up. This allowed a new system to be developed where users can sign content using their own attributes. Since IRMA allows for a practical implementation of attribute-based signatures, it functions as a suitable basis to develop a new toolset. More details on IRMA will be discussed in chapter 5 'Introduction to IRMA'.

4.3.1 Informational Value

Q2: *”How to increase the informational value of digital signatures?”*

The potential advantage of using (IRMA) attribute-based signatures is to increase the informational value of a digital signature. We have seen the advantage of giving users the control on whether or not they want to disclose certain attributes, but we have not yet focused on the side of the receivers (verifier), those who want to verify a signature. The receiver of an attribute-based signature can extract more information about the sender’s (signer) identity when the role of the signer is clear and visible on the signature. By giving the signer the ability to attach specific (personal) attributes to a signature, the informational value of the signature increases. The receiver of this signature can see what the identity of the signer is in a more structured and comprehensible manner.

An example of such a signature where no attributes are involved could be ”I am Alice and I signed this content with my private key, the certificate is delivered by the trusted authority X”. This type of signature only shows the receiver the general (first)name of the signer, the confirmation that the signature is signed (with a certain private key), and what certificate is attached by a trusted authority to guarantee the validity of the signature. The (general) name of a signer does, in most cases, not provide enough identifying information to the receiver. At the same time, the name of the certificate authority does not ring a bell by most receivers. All in all, this approach does not deliver a great amount of information to the receiver, certainly not about the identity of the signer and the source of the content.



Signed by: Alice

Issued by: Certificate Authority X

Date: 01-01-2021 12:00:00

Signed by: Alice
Age over 18: Yes
Nationality: Dutch
Profession: Doctor
Employer: Radboud UMC

Date: 01-01-2021 12:00:00

Figure 1: Simplified signature comparison (Standard vs Attribute-Based)

An alternative approach is the introduction of attributes. Attribute-based signatures allow the signer to attach (personal) attributes to the signature, and thereby disclosing more of their identity in a comprehensible manner while still preserving the legal obligation of data minimization. Furthermore, attribute-based signatures still guarantee the same security properties as ‘standard’ digital signatures. Figure 1 shows a (simplified) comparison between a ‘standard’ signature and an attribute-based signature in terms of what information is displayed. An increase in the amount of (identifying) information and informational value of digital signatures should ultimately lead to making the signatures more comprehensible and more likely to be part of day-to-day usage. If the use of digital signatures becomes part of the standard procedure when publishing digital content, the chance of disinformation having a negative impact in general is reduced.

An example of an attribute-based signature could be ”I am Alice, I am over 18 years of age, I have the Dutch nationality, I am a Doctor, and I work for the Radboud UMC (Radboudumc), and I signed this content with my private key”. This attribute-based signature gives, apart from more control on revealing the signer’s identity and the source of the content, a more comprehensible signature for the receiver to evaluate. Precisely this functionality is what the IRMA project is trying to achieve and already does to a certain extent. Therefore IRMA seems to be a promising candidate to increase the informational value of digital signatures and achieve comprehensible digital signature semantics. Note that the issuer of an attribute is relevant for the level of trust warranted. This means that e.g. an attribute of a Doctors diploma should only be given out by an issuer who is ‘licensed’ to do so, like a university. This same university should however not be trusted to give out attributes like a driver’s license.

5 Introduction to IRMA

Q3: *"What role can IRMA play in realizing the toolset?"*

IRMA is an identity platform developed and maintained by the non-profit foundation *Privacy by Design.* As the acronym of IRMA suggests, the goal of IRMA is to give users more control over their privacy by revealing (personal) attributes. IRMA enables users to perform two actions namely, (1) selectively disclose certain (personal) attributes to digitally authenticate themselves, and (2) sign digital content using IRMA's attribute-based signatures. The users can use their mobile phone in combination with the IRMA mobile app[17] to perform these actions. Privacy protection is a core part of IRMA, conforming to the privacy by design requirements stated within the GDPR [Cou16], which is mandatory for new ICT-systems within the European Union. But IRMA goes further than the mandatory privacy by design principles. The system also provides protection against notions like identity-fraud, linkability of the user's activities, and giving the user a view on who requests what data. More details about the privacy friendliness of IRMA will be discussed in chapter 5.3 'privacy friendliness'.

5.1 Privacy by Design Principles

The Privacy by Design Foundation aims to develop open and privacy-friendly digital solutions. Currently, the main project within the foundation is IRMA. The aim of the IRMA project was to develop a digital implementation of privacy-friendly authentication based on attribute-based credentials. The project further extended to signing digital content with an attribute-based signature. Both these sub-projects within IRMA are developed with the privacy by design doctrine principles in mind. The privacy by design principles were introduced by Cavoukian [Cav09] with the general idea that the system should be designed in a way that avoids, or at least minimizes, the amount of personal data processed. In other words, satisfying *data minimization.* The key elements of data minimization are: *Anonymization*, *separation*, and *pseudonyms*:

- The anonymization and / or deletion of personal data as soon as possible,
- separation of user identity,
- and the use of pseudonyms where possible.

These elements combined are what form the principles of privacy by design, and by which the Privacy by Design Foundation develops open and privacy-friendly digital solutions, like IRMA.

5.2 Current Developments

The software developed by the Privacy by Design Foundation is open source and available on GitHub[18]. Making the software open source gives certain advantages like reliability validation and software contributions by those who are interested. Bearing no

[17] https://irma.app/docs/irma-app/

[18] https://github.com/privacybydesign

secrets and functioning on a transparent playing field. It supports the overall idea of making IRMA available to the public, where the project should ultimately be picked up and supported by national institutions worldwide.

Several projects (and demos) have been, and still are, in development using IRMA as its basis. Projects like IRMATube[19], IRMA-meet[20], IRMA-vote[21], and the fundamental project IRMA-app[22]. All of these projects are currently developed and maintained by the Privacy by Design Foundation, supported by SIDN[23]. IRMATube allows users to watch videos when they disclose certain attributes, like disclosing your age when the movie or video is age-restricted. IRMA-vote can be used for online (anonymous) voting. And IRMA-meet enables users to initiate or join a video call by disclosing certain attributes. This would often be your name, but it could be any relevant attribute that the video call host would require of the participants. In order to disclose these attributes, the IRMA-app (mobile app) is always required. The app stores your attributes and handles the issuing and disclosure sessions with the issuer and service provider, respectively. IRMA is internationally available, as can be seen in the IRMA dashboard[24]. This corresponds with the mission of IRMA to be integrated within society. Making people all over the world conscious of the existence of IRMA and the potential benefits it can have on their level of privacy.

5.3 Privacy Friendliness

IRMA claims to increase the user's control over their privacy, consequently being privacy-friendly and privacy-preserving. We have already seen the privacy by design principles that are used as a basis for the development of IRMA. Other relevant privacy-preserving properties are *data-minimization*, *identity-fraud*, *linkability*, and *decentralization.*

5.3.1 Data-Minimization

Another notion related to IRMA is *contextual authentication* [JS20]. When a service provider asks the user to reveal certain attributes necessary and relevant to the provided service, the user has the decision to accept the request and disclose his attributes (or not) using the IRMA mobile app. This conforms with the GDPR's data minimization requirements.

Data minimization functions from two ways. On one side is the user of the IRMA mobile app. Users can selectively disclose attributes, which enables the ability for the user to minimize the sharing of his data. And on the other side is the service provider. A service provider needs certain attributes from the user to deliver the requested service. To conform to the requirements of data minimization, the service provider should only request the necessary attributes. For the user, the selective disclosure is, on average, more appealing than to a service provider. Service providers can often use the (personal) data of the user to improve their business processes or more lucrative activities like generating a more efficient revenue stream using targeted ads.

[19] IRMATube

[20] IRMA-meet

[21] https://www.ru.nl/ihub/research/research-projects/irma-vote/

[22] IRMA-app

[23] https://www.sidn.nl/

[24] IRMA Dashboard

5.3.2 Tracking and Decentralization

Some of the concerns regarding systems that claim to be privacy friendliness are the inclusion of tracking cookies and the use of a central database. IRMA has neither. There is no tracking of any activity by IRMA or the Privacy by Design foundation. This means that whenever a user discloses attributes or performs some action within the IRMA mobile app, the data is exclusively exchanged between the IRMA mobile app and the service provider. There is no intermediate third-party functioning as a privacy hotspot. There is also less intention to do this since the Privacy by Design foundation is set up to be non-profit, and therefore not interested in tracking the activity of users to improve their revenue stream. The other concern relates to the use of a central database. Often (mobile) applications make use of a central database that is under the control of the organization that develops the application. The IRMA mobile app is different since it does not require a central database to store the user's attributes. All credentials and related attributes are stored locally within the IRMA mobile app. The app is therefore responsible for the security of the attributes, not a central database. The advantage to this decentralized approach is that no third party, including the Privacy by Design foundation, can see any of the attributes belonging to a certain user, in other words implementing *unlinkability*. Note that there is a central database required to prove half of the secret key, more details on the functioning of the secret key will be discussed in chapter 6.2.1 'Special Attributes'. Unlinkability and zero-knowledge proofs are discussed in detail at chapter 6.1.2 'Zero-knowledge proofs'. Figure 2 comes from the website of the Privacy by Design Foundation and portrays the difference between a decentralized (IRMA) and centralized (non-IRMA) architecture.

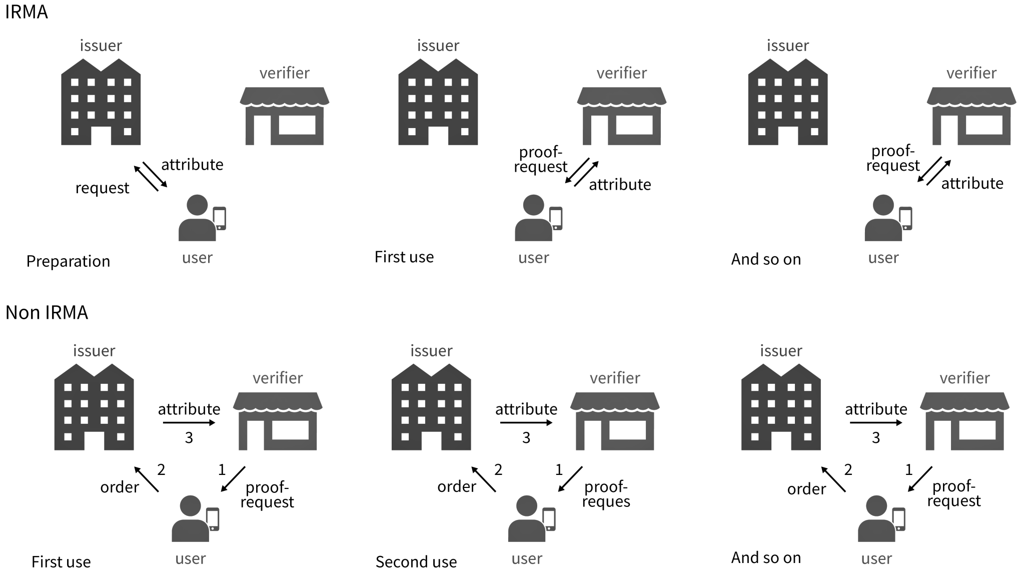


Figure 2: Decentralized vs. Centralized Architecture.

In a centralized (non-IRMA) architecture the user has to make a request to the issuer each time he wants to prove some attribute(s). This means that a third-party (the issuer) is always required to communicate with a verifier. This third-party is therefore capable to 'monitor' the requests of the user, which is not beneficial to the user's privacy. This is solved by decentralizing (IRMA) the architecture. The user requests an attribute once from the issuer, from this point communication with a third-party (issuer) is not required anymore, unless the user want to receive another attribute. The user stores the requested attribute(s) in the IRMA mobile app, and can disclose the attribute(s) directly to a verifier when requested.

5.4 Legal Status of IRMA Signatures

The eIDAS defines three levels of authentication: *low*, *substantial*, and *high*. These three levels determine the security assurance of authentication techniques. Regarding IRMA authentication, it is somewhat harder to define in which level it fits. It can be stated that the IRMA authentication is 'naive'. The security assurance that can be given is determined by combining a specific IRMA credential (containing attributes) and the security assurance level of the specific attributes. There is no unambiguous eIDAS level when looking at IRMA as a whole.

When looking at the eIDAS regulation regarding digital signatures, as mentioned in chapter 4.2 'eIDAS Regulation', we see that there are again three levels: *simple*, *advanced*, and *qualified*. The eIDAS regulation, as of writing this thesis, only accepts digital signatures to 'claim' the level of 'qualified' when there is a qualified certificate involved. IRMA does not have this functionality since it aims to exclude third parties like certificate authorities who can distribute qualified certificates. Therefore, no qualified certificate is available to base the signature on. IRMA's attribute-based signatures can claim the level of 'advanced'. However, the ability to claim this eIDAS level depends on the security and reliability level of the IRMA credentials.

5.5 Role of IRMA

As we have seen, IRMA provides several benefits when it comes to privacy-friendliness of authentication and signage processes. Since this thesis researches the potential of increased comprehensibility of digital signature semantics using attribute-based signatures, the focus will be on the signage part of IRMA. The role IRMA can play is forming the basis for the foreseen toolset. The toolset can take advantage of the technologies that IRMA delivers regarding the creation of attribute-based signatures and the validation of digital content signed with an attribute-based signature. Future developments within IRMA can also be translated into the toolset when relevant.

So, IRMA can be used as a basis for realizing a practical implementation of the attribute-based signature toolset. Theoretically, the use of attributes to make digital signatures more comprehensible seems promising. However, we need a way to make this practical. The upcoming chapters will look into how we can transform this theoretical solution into a practical solution using IRMA. Making digital signatures both comprehensible and accessible to the public.

6 Technical Specifications of IRMA

Q4: *"What are the technical requirements and related security properties when realizing the toolset?"*

The IRMA system is used as a basis to develop the toolset. We have seen the theoretical possibilities of using IRMA but have yet to explore what is possible in the practical sense. In this chapter, we will see the technical specifications that are present within IRMA and describe how each of the specifications functions. This includes, among others things, the participants that are necessary for the IRMA system to be successful and which actions are available for the user to perform when using an IRMA-based application or mobile app. Finally, we will discuss the functioning of both the authentication and the signature schemes present within the IRMA system.

6.1 Sessions

Within IRMA multiple session requests can be made, including the *disclosure* request, *issuance* request, *combined issuance-disclosure* request, and *attribute-based signature* request. IRMA allows different so called *participants* to communicate with each other by making requests and completing IRMA sessions. Participants involved during an IRMA session are:

- **User**: a person who wants more control over his or her privacy.
- **Verifier**: a party to which a user authenticates himself. The party verifies the user's attributes to provide a certain service. Therefore, this party is sometimes referred to as *Service Provider.*
- **Issuer**: verified organizations capable of issuing attributes to users using an Idemix private key, sometimes referred to as *Identity Provider.*
- **Requestor**: a party that can take the role of both a service provider (verifier) or identity provider (issuer). A requestor can issue attributes to the user, verify attributes and attribute-based signatures, and / or sign digital content with an attribute-based signature.
- **Scheme Manager**: determines and distributes Idemix public keys, credential types and issuer information (to requestors). The scheme manager also decides what credential types issuers may issue and which issuers may join the scheme manager's domain.
- **Mobile App**: can both receive and disclose attributes, and store the attributes within cards (credentials). The mobile app acts as the *client* in the IRMA protocol.

Users can, apart from gaining increased control over their privacy, disclose their attributes to authenticate themselves or sign certain digital content. Therefore, the user needs a *requestor* to communicate with. The requestor is often realized in the form of a web or desktop application. The requestor makes use of the backend[25] and frontend[26] packages delivered by the open-source software from the IRMA project. This allows the

[25] https://irma.app/docs/irma-backend/

[26] https://irma.app/docs/irma-frontend/

requestor to talk to an IRMA *server*. The requestor initiates a session with the server, followed by the server initiating a session with the IRMA *mobile app*. The requestor will show a QR code for the user to scan, followed by the user entering a PIN code and confirming the request on the mobile app, and thereby completing the IRMA session.

Figure 3 shows the IRMA session flow in a visually detailed representation[27].

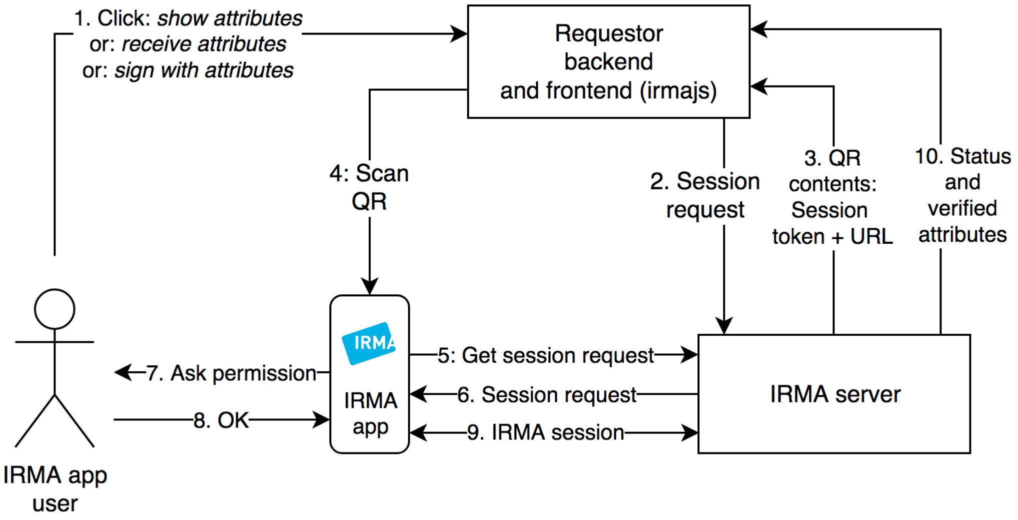


Figure 3: IRMA issue (receive), disclose (show), and sign session.

The IRMA server handles the IRMA protocol and ensures the session between the mobile app and the requestor. The IRMA mobile app, previously known as an 'IRMA Token' (in older documentation), can perform three different session types with the IRMA server:

- **Issuance sessions**: a new set of IRMA attributes is received by the IRMA mobile app, including the issuer signature from the IRMA server. The signature is used for future attribute-based signature and disclosure sessions. Before receiving a new set of IRMA attributes, it is possible to ask the user to disclose some attributes which are already stored within the IRMA mobile app. This session type is called the 'combined issuance-disclosure session'.
- **Disclosure sessions**: this session allows users to disclose specific attributes requested by the requestor. The user can make a disclosure request, using the requestor, to the IRMA server. This initiates a disclosure session. The IRMA server sends the disclosure request to the IRMA mobile app. If the mobile app user confirms the request, the disclosed attributes are sent back to the IRMA server. Finally, the IRMA server will verify the disclosed attributes and sends its validity status, including the verified disclosed attributes, back to the requestor.
- **Attribute-based signature sessions**: this session allows a user to attach an IRMA attribute-based signature to a message. Fundamentally it functions the same as any other digital signature. However, the advantage over 'basic' digital signature is the ability to attach (personal) attributes while signing a message.

[27] https://irma.app/docs/assets/irmaflow.png

This not only benefits the signer, in terms of increased privacy control, but also the verifier. The verifier, in this case those who read the message, receives useful information about the singer in the form of (personal) attributes. Attributes are attached to a message digitally signed into an IRMA attribute-based signature. Verifying the IRMA attribute-based signature can be done at any (future) date. The signature guarantees that the message has not been altered and that the IRMA attributes attached to the signature were valid at the time of creating the signature. Note that in the current version of the IRMA attribute-based signature only messages in the form of a string can be signed.

6.1.1 Keyshare Protocol

Before certain IRMA sessions can proceed, the mobile app may require the user to enter his PIN code. This allows the requestor to be more certain about the fact that the user, who initiates the session, truly owns the attribute. Preventing malicious activities like a malicious person using a stolen phone to disclose attributes (present on the stolen phone) that do not belong to him. The *IRMA keyshare server* bears the responsibility of verifying the correctness of the PIN code. In case the PIN code is incorrect, the session must be aborted. The keyshare server communicates with the mobile app and possibly with the IRMA API server in a protocol called the *keyshare protocol*.

Before any IRMA session can occur the IRMA scheme, managed by the scheme manager, must employ a keyshare server. The keyshare server is involved in any IRMA session that involves attributes which are under the responsibility of the scheme manager. IRMA users register to keyshare servers of the installed scheme managers when the IRMA mobile app is installed and opened for the first time. This is also the moment that the user can choose the PIN code. From there, each time the user initiates an IRMA session, the PIN code is required before the keyshare servers allow the session to be successfully completed.

The keyshare server holds a few responsibilities. (1) No value of any attribute is learned by the keyshare server. (2) The revocation of half of the private key by the user. Meaning that whenever the user's phone is lost (or stolen), it should be possible for the user to remotely block their IRMA mobile app from executing any future IRMA sessions. (3) Before an IRMA session occurs, there must be a check that the user initiating the session is the same user that registered to the keyshare server before. If the user initiating the session is not the same as the user registered to the keyshare server, the session must be blocked.

The cryptography used in the IRMA sessions is implemented in such a way that a keyshare server must be involved to complete a session. This ensures that the keyshare server can reliably block sessions from being executed by refusing to cooperate. This approach is necessary because the first two responsibilities of the keyshare server expose that verifying the PIN code locally in the mobile app is not sufficiently secure. The mobile app could be altered, creating a malicious version of the app that does not validate the PIN code.

6.1.2 Zero-knowledge proofs

To perform a disclosure session, a *zero-knowledge proof*[28] is required. Both the IRMA secret key and signature are kept hidden from the verifier using a zero-knowledge proof. The signature is kept hidden to ensure unlinkability of the credentials. Furthermore, zero-knowledge proofs prove that a number satisfies a certain property without disclosing the number itself. Within a credential, there can be multiple attributes. When a user discloses only a certain set of attributes within a credential, the other attributes (including the secret key) are hidden using the zero-knowledge proof. This way, the user can convince the verifier that he has a valid issuer signature over all attributes within the related credential, including the hidden attributes.

Additionally, we have the issuers who want to safely sign the user's attributes without knowing the user's secret key. The signing of the user's attribute by issuers is needed to later on prove that certain attributes are provided by certain (authentic) issuers, linked to a specific user. To keep the user's secret key hidden (and thereby private) a zero-knowledge proof is used. The first attribute of each received credential is always the user's secret key. The user proves to the issuer that he knows the first attribute (secret key) using a zero-knowledge proof. This way the issuer knows its the correct user, without the user's secret key being disclosed to the issuer. The issuer can then safely sign the attributes.

6.2 Attributes & Credentials

An *attribute* is a cryptographic entity resembling a statement or property about a person such as age, profession, or name. Each specific attribute tells something about you as a person. The property of an attribute can either be *identifying* or *non-identifying.* If you are a doctor, one of your identifying attributes can be 'I am a Doctor'. And if this doctor is over the age of 50, an additional non-identifying attribute can be 'I am over the age of 50'. The first attribute tells something about the person's identity, while the second does not uniquely identify a person. In addition to this property, each attribute is authentic. Attributes are grouped into a cryptographic container termed as a *credential.* As mentioned earlier, IRMA uses the Idemix attribute-based credential scheme that extends the idea of using credentials. This scheme allows credentials to be issued to a user by a trusted party, mainly official authorities, called the issuer. Equal to the issuer participant within IRMA. The issuer creates a digital signature over the credential using its private key. The user will receive both the issuer's signature as well as the credentials containing grouped attributes. Furthermore, the Idemix attribute-based credential scheme provides an *attribute disclosure protocol* in which the user can selectively disclose any subset of attributes to the verifier. More on selective disclosure will be discussed in chapter 6.4 'Selective Disclosure'.

[28] IRMA's implementation of Zero-Knowledge Proofs https://irma.app/docs/zkp/

6.2.1 Special Attributes

Within each credential there are ‘special’ attributes: the *metadata* attribute and the *secret key* attribute. The metadata attribute is always present as a special attribute within each credential and is always disclosed. In figure 4 an IRMA credential is illustrated. The metadata attribute gives information about the date at which the related credential was issued, which *credential type* the related credential is an instance of, and the expiry date of the related credential.

The *secret key* attribute, called the user’s secret key, is the first attribute of any credential. Hence, this attribute is, as the metadata attribute, always present within a credential. This secret key is a randomly chosen 256-bit integer and partly stored by the user’s IRMA mobile app when it is run for the first time. Note that the other half of the secret key is stored on the keyshare server. The app makes sure whenever a new credential is received, this number (secret key) is used as the first attribute. Consequently, all credentials stored on the mobile app will have the same first attribute, the secret key. Where the metadata attribute is always disclosed, the secret key attribute is never disclosed. Meaning that this attribute is always kept hidden, even from the issuer during issuance. The secret key is only known to the user.

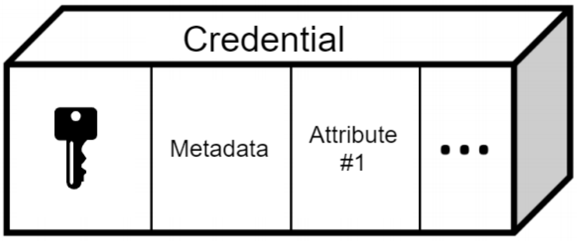


Figure 4: IRMA credential.

The secret key guarantees two properties. (1) It enforces the control of the user. The user can only disclose attributes from a credential when it knows the value of all attributes within that specific credential, including the value of the secret key. And since the user is the only one who can know the secret key, the participation of the user himself is guaranteed in the disclosure session. (2) When attributes of multiple credentials are combined during disclosure the secret key is also of importance. Attributes can be disclosed from multiple credentials at the same time. The disclosed attributes from the different credentials should belong to the same user, otherwise a user can create a fake identity by combining credentials of other users. Therefore, the verifier checks whether the secret key of each credential, included in the disclosure session, is the same and corresponds with the secret key of the user.

In practice the second property results in the following. In case the user is disclosing attributes from two or more different credentials the proof of knowledge, which the IRMA mobile app calculates and sends to the requestor, comes into play. Both the issuer signature linked to each credential must be valid, and the first attribute of all credentials must be corresponding. The requestor can now be sure that the user is not ‘pooling’ credentials, but instead only disclosing attributes associated to a unique person.

6.2.2 Credential Types

Each credential is an instance of a *credential type.* Credential types define specific properties of a credential like what their name is, the related issuer, and the number of attributes within the credential. Another optional credential type, defined by the *scheme manager*, is a *singleton.* In case this 'singleton' property is present, the IRMA mobile app will store at most one instance of this specific credential type. When a new instance is received, the old instance will be overwritten. An attribute that is part of a credential type marked as a singleton could e.g. be a unique personal identification number. The user can have multiple instances if the credential type is not a singleton. Non-singleton credentials are e.g. bank account details, e-mail addresses, diplomas, mobile phone numbers, and many more.

6.3 Schemes

Every participant within IRMA must be aware of the existing attribute names, credential types, and issuers, including the public keys of the issuers. This information is contained within IRMA *schemes*, managed by the *scheme manager.* The scheme manager distributes the information to all parties, defaults of such schemes are provided by the Privacy by Design foundation in the form of a directory structure[29]. The information provided by the scheme manager consists of (1) all credential types the specific issuer can issue. (2) All the different issuers that use this scheme. And (3) the Idemix public keys of the relevant issuers. The scheme manager has an Elliptic Curve Digital Signature Algorithm (ECDSA) [JMV01] private-public key pair by which the directory structure is signed. All the information within a scheme is disturbed and signed by the scheme manager. Meaning that the scheme manager has full and exclusive control over which issuer can use its scheme, including which credential types and attributes the issuer can issue. Note that anyone can create their own IRMA scheme.

6.4 Selective Disclosure

A core feature of the Idemix technology is *selective disclosure.* Selective disclosure enables users to control which attributes will be disclosed to the service provider (or verifier). When requesting a service from the service provider, the user needs to disclose certain attributes. This requires the user to provide credentials that contain the related attributes. However, the user may choose to reveal only a subset of all attributes contained in the credential. This selective disclosure of attributes gives users greater control over their privacy since only necessary attributes chosen by the user are disclosed.

The user can prove that he is truly the rightful owner of the credentials since the credentials are signed by the issuer. In the issuance procedure of new attributes, the issuer and the user work together to create a new credential. The user has to authenticate to the issuer, followed by the issuer collecting attributes for the user. The issuer and user then jointly perform a cryptographic protocol in which the attributes are combined into a fresh credential. This credential then gets signed by the issuer. The freshly created credential contains the relevant attributes and the user's secret key.

[29] Github: pbdf-schememanager and Github: irma-demo-schememanager

A user can have multiple credentials, each containing a different set of attributes. The user can decide to reveal a full credential to the service provider or only a subset of attributes within that credential. So, in case the user wants to sign digital content using IRMA attribute-based signatures, the attributes can be selectively disclosed. Later we will see the IRMA Signature Application (ISA) as a requestor that enables the user to sign digital content using an IRMA attribute-based signature. The user is allowed to select specific attributes to be included within the signature.

6.4.1 Authentication and Signature Schemes

The IRMA systems consists of two schemes, *authentication* and *signature*. Where in the authentication scheme attributes can be selectively disclosed by the user when authenticating themselves to a certain service provider. And where in the signature scheme the user can selectively disclose and attach attributes to an attribute-based signature. The selective disclosure proof is used either for authentication (with a fresh nonce as input) or for signature generation (with the hash of a message as input).

During an IRMA authentication, the verifier sends a nonce to the IRMA mobile app to be included in the selective disclosure (SD) proof generation. This nonce is strongly bound to the proof and it helps the verifier check the freshness of the proof. This should prevent a user from replaying the same proof to authenticate during different authentication sessions. Replay attacks will be discussed in more detail at chapter 7.2 'Replay Attacks'.

'Towards practical attribute-based signatures' by Hampiholi et. al. [HAvdBJ15] proposed to adapt this (authentication) approach into the IRMA attribute-based signature: "If the hash of a message is used during a SD proof generation instead of the nonce, then the SD proof becomes the user's signature on the message. So, the main functional difference between a SD proof in IRMA authentication and IRMA signatures is the way the nonce is defined." This new approach of using the hash of a message is now part of the selective disclosure protocol for IRMA attribute-based signatures, see figure 5.

The IRMA attribute-based signature scheme consists of four algorithms: *Key Generation*, *Attribute Issuance*, *Signature Generation*, and *Signature Verification*. Note that the first two algorithms are also used for the IRMA authentication scheme. Algorithm 3 and 4 are shown in figure 5.

1. *Key Generation*: when the mobile app is initialized (installed and run for the first time by the user), a secret key is generated and stored securely. This secret key is used during issuance of attributes, authentication when requesting attributes, and when signing some digital content.

2. *Attribute Issuance*: the user can obtain attributes from authorized issuers. The issuer signs the credential containing the requested attributes with its private key. The corresponding public key is used by the verifiers when validating a signature.

3. *Signature Generation*: the mobile app is responsible for the generation of the signature. The mobile app outputs the required attributes, timestamp and the selective disclosure proof ensuring that the mobile app user has signed the message and possesses the correct attributes issued by the issuer.

4. *Signature Verification*: verifiers who want to validate an attribute-based signature need to have the public key(s) of the issuer(s) that issued the related attributes to the mobile app. The issuer's public key is used for validating the signature. The verifier checks whether the timestamp is valid and whether the message (within the signature) was signed by the user of the mobile app who possesses the related attributes issued by an (authorized) issuer.

Details about the timestamps will be discussed in chapter 7.5.2 'Timestamps'.

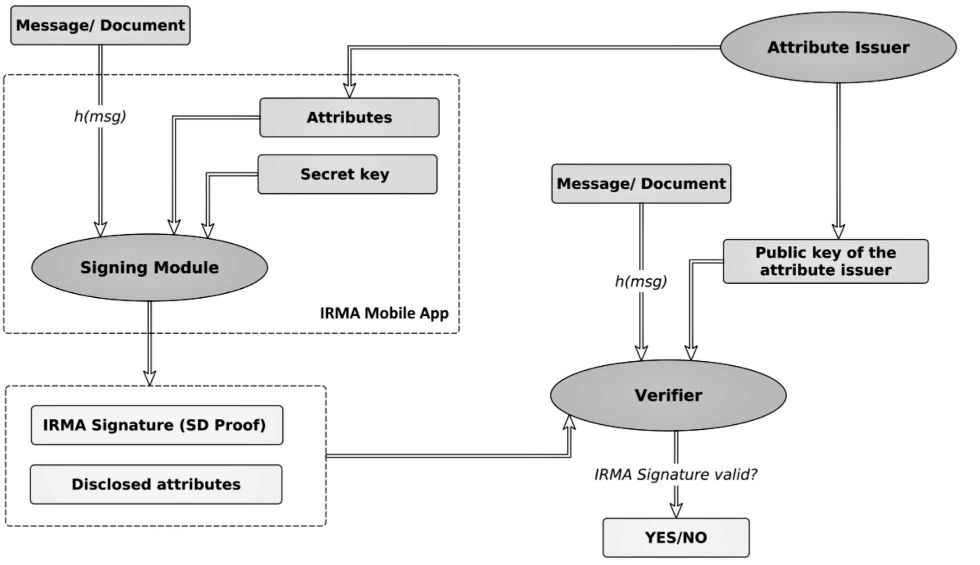


Figure 5: IRMA attribute-based signature scheme (without timestamps) by Hampiholi et. al. [HAvdBJ15].

7 Security Properties and Guarantees

In this chapter, the security properties and guarantees related to disclosure and issuing of attributes and credentials (mainly relevant to the authentication sessions that users have with service providers) are discussed first. The IRMA system originated with the IRMA authentication which, in terms of security, mainly relates to the disclosing and issuing of attributes. Based on the authentication scheme, the IRMA signature was developed. So, after discussing the security properties and guarantees of disclosure and issuing, we will look into the (additional) security properties and guarantees of signing digital content using IRMA's attribute-based signatures. Finally, we will discuss the current limitations and general societal issues present when using IRMA as a basis of the toolset.

7.1 Revocation

When an IRMA disclosure session is performed we need to be certain about the validity of the user's credential and its containing attributes. An attribute like 'I am a doctor' should only be valid as long as the user is officially a doctor. The user who is trying to disclose an invalid attribute, loaded within a credential on the mobile app, should not be able to disclose this specific attribute until it is reissued. Therefore we do need some mechanism to revoke attributes. Within IRMA this mechanism is conveniently called *revocation*. Note that the revocation mechanism is currently in an experimental stage.[30]

Issuers can revoke a credential that is previously issued to a mobile app. Issuers can revoke a credential when one or more contained attributes are no longer valid. Within scientific research related to revocation, there seems to be a consensus to define parties who are capable of revoking credentials as the *revocation authority*, in IRMA's case the revocation authorities are the issuers. Parties who want to verify attributes (verifiers) can establish whether or not the received attributes are still accurate by requesting a proof of nonrevocation from the mobile app. However, the verifier relies on the issuer (revocation authority) to revoke invalid attributes, including the related credential. Verifiers can request nonrevocation proof for a given credential type by including it in a revocation array which can be a part of the disclosure session request.

Revocation in IRMA is an implementation of the RSA-B scheme by Baldimtsi et. al. [BCD$^+$17]. The scheme allows the mobile app to prove nonrevocation of credentials using a zero-knowledge algorithm. This guarantees that multiple disclosures of identical attributes within a credential are unlinkable. When two IRMA sessions are performed, where the to-be-disclosed attributes are identical, no party should be able to identify if the session is performed by one user or two distinct users. Therefore these types of sessions are not linkable as coming from the same user. The IRMA documentation states that this security property only holds at the cryptographic level, meaning that on the transport layer e.g. the user's IP address, the verifying party could potentially still link sessions. Note that revocation is enabled per credential type in the IRMA scheme.

[30] https://irma.app/docs/revocation/

7.2 Replay Attacks

It should be impossible for eavesdroppers to replay an IRMA disclosure session. When attributes are verified, a random number of bits, called the nonce, is sent by the verifier to the mobile app. The mobile app responds with the disclosed attributes and the proofs of knowledge that fit on this nonce in a precise manner. Note that the verifier cannot reuse nonces, since this would break the security against replay attacks.

Verifiers should also not be able to perform replay attacks. To ensure this from happening, the IRMA app never sends a complete copy of the credential's signature to the verifier. Instead, parts of the signature are hidden using proofs of knowledge, ensuring unlinkability. This way verifiers cannot exploit what they learn in an IRMA dislosure session. E.g. disclose attributes, received from a user, to other verifiers, and start acting as an IRMA mobile app.

7.3 False-Identity

Since users only disclose relevant (personal) attributes, the ability to create a full profile of someone and use this profile to perform identity-fraud becomes difficult, perhaps even impossible. When certain attributes are never disclosed, they cannot be exploited for identity-fraud. Imagine a malicious *service provider* that asks the user to authenticate himself with his attributes. The malicious service provider could request an excess of (personal) information which is not required to deliver the service. The goal of the malicious service provider is to extract as much (personal) information from the user as possible. Technically this is possible, however the user still has to agree to this request. The user could simply decline the request, and no attributes will be disclosed. Leaving the malicious service provider with empty hands. Though, it should be clear to the user which attributes are requested. If this would not be clear, the user could accidentally agree to the request.

Additionally, credentials containing attributes within the IRMA mobile app are digitally signed by the issuer. This means that a malicious *user* is unable to create a 'fake' attribute. However, the issuer must be trusted and should therefore only be able to function as an issuer when validated, like licensed organizations or authorities.

7.4 Security Guarantees

Apart from revocation, replay attacks, and false-identity IRMA also provides the following security guarantees on the user's credentials when disclosing or issuing attributes, partly based on the analysis of Vullers & Alpár [VA13]:

- *Integrity and Authenticity*: it should be impossible to create or alter a (valid) credential without cooperation between both the user and the issuer. Only valid credentials can be issued by the credential's issuer. The credential exclusively belongs to a unique user.
- *Multi-show unlinkability*: it should be impossible for a verifier to link two IRMA sessions to each other. So, when a certain credential is shown multiple times to a verifier, there is no way to link the different sessions. This can only be guaranteed

when the disclosed attribute cannot be uniquely identified by the user. If this is not the case the information could potentially be used to link two IRMA sessions.

- *Issuer unlinkability*: it should impossible for the issuer to link usage of its own credentials in a verification process to a specific issuing session. The issuer cannot link the credential to the user, when this credential is used. This guarantees that profiling of credentials is not possible.
- *Credential unforgeability*: only issuers, who hold an Idemix private issuance key, can issue credentials. The issuer's Idemix private key is used to sign credentials using a CL (Camenisch-Lysyanskaya) signature [CL02]. A CL signature guarantees the integrity of the credentials i.e. adding, modification, and deleting of attributes to a credential can be detected by a verifier. Resulting in a guarantee to the verifiers that the received attributes are valid, and were issued by a certain issuer.
- *Non-Transferability*: it should be impossible to transfer credentials from one user to another, even if both users agree to this. To ensure that this transfer does not occur the secret key is used, as mentioned before. The secret key cannot be copied and used by different users.

7.5 Signature Security

In terms of security an IRMA attribute-based signature guarantees the integrity, authenticity, and time of the related digital content. The digital content cannot be altered without notice (integrity), and the attributes and secret key are bound to the issuance and the signature (authenticity). These security properties are no different from the security properties of a 'basic' digital signature. In chapter 6.4.1 'Authentication and Signature Schemes' we have seen the IRMA attribute-based signature scheme, showing the process of creating an IRMA attribute-based signature. The following chapters will discuss relevant security properties, specifically related to IRMA's attribute-based signatures.

7.5.1 Revocation

When digital content is signed using an IRMA attribute-based signature, we want to be certain about the accuracy of a credential and its containing attributes. As we have seen in chapter 7.1 'Revocation' IRMA authentication already has a revocation scheme. It adds the ability for issuers to revoke a credential that it previously issued to a user, more specifically to an IRMA mobile app. There is currently no design of a revocation scheme for digital signatures. However, ones there is such a scheme it should allow requestors, those who verify a signature, to use the scheme as part of the signature validation. When a user signs digital content using an attribute-based signature the attributes that are attached should be accurate, e.g. the user should still be a doctor if he wants to sign digital content using a doctor's profession attribute.

Additionally, the design of a privacy-friendly revocation scheme for digital signatures that ensures complete unlinkability is currently not trivial [HAvdBJ15]. The existing revocation scheme for IRMA authentication, mentioned in chapter 7.1 'Revocation', has to be adapted to IRMA attribute-based signatures. As a possible solution for the lacking revocation scheme regarding signatures, the attribute expiry dates can be made shorter while concurrently the re-issuing of attributes is made simpler. This way attributes are ‘revoked’ (by expiring) regularly, while re-issuing the attributes is easy and quick for the user.

7.5.2 Timestamps

The timestamp security property of IRMA’s attribute-based signatures, and digital signatures in general, is determining the actual time of signing. The function of a timestamp on a digital signature is to determine at what specific date and time the content was signed. This guarantees that the signed content existed at a point-in-time and has not been altered since. Within IRMA there are two types of timestamps, (1) the date and time at which a digital signature was generated and (2) the expiry dates of the attributes attached to the digital signature. The signer can obtain an authorized timestamp signed by a Timestamp Authority (TA). The *proof status* of a timestamp is part of the IRMA signature request and can result in the following:

- *Signature generation date.* For the verification process of an attribute-based signature to be successful, it has to include a valid timestamp. If the timestamp is invalid, the proof status will give an ”INVALID_TIMESTAMP” response.
- *Attributes expiry date.* Each attribute has an expiry date, and since attributes should be valid when generating an attribute-based signature, the expiry date should be greater than the time present within the timestamp. Therefore, when a selective disclosure proof is performed the validity of the timestamp should be checked. If one or more attributes were expired at the time of creating the attribute-based signature, the proof status would give an ”EXPIRED” response.

During an attribute-based signature session, the IRMA mobile app will sign the message and validate whether the user’s attributes are valid, including the related timestamp (expiry date). This validation happens within the IRMA mobile app before the message is signed. If at least one of the attributes contains an invalid timestamp the message will not be signed. Note that an IRMA server is required to perform the attribute-based signature session. The server handles IRMA-specific cryptographic details related to the session with an IRMA app on behalf of a requestor. The requestor is in this case the application that wants to sign digital content.

To prove whether the signature timestamp is (in)valid a requestor can integrate the *IRMA package*[31] (or module). The IRMA package contains generic IRMA structs and logic of use to all IRMA participants. Most relevant to this thesis is the attribute, credential, and signature verification logic. A user can submit (attribute-based) signed digital content to the requestor, followed by the requestor using the signature verification logic to verify whether the attached signature is valid. Once the signature is verified, the requestor can display the proof status (valid or invalid) to the user.

When signing digital content using an attribute-based signature the related attributes should be valid at the time of signing. Attribute validation in combination with the timestamp guarantees (to the verifier) that the attributes attached to the signature were valid at the time of signing. The attributes can become invalid at a later time. E.g. when the signer of some digital content uses his profession as an attribute, but a few days later he is enjoying his retirement. At the moment of signing he was entitled to sign

[31] https://pkg.go.dev/github.com/privacybydesign/irmago

using his profession attribute, but that same attribute is invalid when he retires. When someone is verifying the signature he at least knows that the attached attributes are valid at the time of signing. If the verifier want to have an up-to-date signature he would need to request a new one from the signer. Note that 'attribute validation' is currently an abstract requirements since it is not possible within IRMA to validate and revoke an attribute within the signature scheme, as discussed in chapter 7.5.1 'Revocation'.

7.5.3 Non-re-usability

The signature should not be 'split' from the digital content and then attached to different digital content. In other words, the signature should be *non-re-usable*. In the current version of IRMA the user is only capable of signing a string (combination of characters), therefore an alternative is required when we want to let users sign any type of digital content. We solve this by using the hash-and-sign scheme as discussed in detail later in chapter 7.6.1 'Hash-and-Sign Scheme'. The digital content is hashed and stored as the 'message' within the IRMA attribute-based signature. Originally the message entity within IRMA signatures allows a simple message (string) as input. By hashing the digital content, e.g. a PDF file, and giving this hash as input to the message entity, we can guarantee non-re-usability. The digital content is encoded to a base64 encoding. This encoding is hashed. The resulting hash is a unique one-way compression of the base64 encoding and converted into a string. If the attribute-based signature were attached to different digital content, the hash of the new digital content would not correspond with the hash of the original document, making the signature invalid. This process of signature validation can be executed by the verifying party. The verifying party can hash the digital content with the attached attribute-based signature, followed by comparing the hash that is present in the message entity of the attached attribute-based signature. If the two hashes are equal, it means that the signature is attached to the original document, and if the two hashes are not equal, than the signature does not belong to the attached signature. Preventing the re-usability of the signature.

7.6 Limitations & General Societal Issues

The use of IRMA as a basis to develop an attribute-based signature toolset comes with a limitation as mentioned before. To solve this limitation, the so-called *Hash-and-Sign scheme* is introduced. This scheme makes it possible for any type of digital content to be signed using an attribute-based signature, instead of only allowing a string to be signed. This scheme is necessary until IRMA implements this scheme, or a similar solution to sign any type of digital content, into the IRMA attribute-based signature. Additionally, two societal issues are discussed to portray why it is inevitable that a certain amount of responsibilities still lies with the user when trying to preserve privacy.

7.6.1 Hash-and-Sign Scheme

In the current version of IRMA (2021), it is impossible to sign or attach an attribute-based signature to a document or any type of file, only to a message (string). This prevents users from being able to e.g. attach the signature to a PDF file. Since the aim of this thesis is not to find a solution to this problem within IRMA, a specific solution for the toolset is introduced called the Hash-and-Sign scheme.

A hash-and-sign scheme works as follows: the document, which can be of any type, is converted to a *base64* encoding. The output of the base64 encoding is *hashed* using the SHA-3 hashing algorithm. The hashing function ensures that the base64 encoding of any file is processed to a fixed-size value. This results in a hash that can be used as input for the 'message' entity within the attribute-based signature. Instead of writing a custom message (string) within the signature, the hash is given as input for the message. Finally, the hash is signed using the attribute-based signature. This approach provides three advantages, (1) the core functionality of signing within the current version of IRMA attribute-based signatures is kept intact. (2) The input can be of any file type, whether it be a PDF, DOC, TXT, or any other file type. And (3) it ensures the notion of non-re-usability, as mentioned before.

7.6.2 Shoulder Surfing

Attributes within the IRMA mobile app are used to create the user's identity, or in other words, create a 'passport' that can be used to selectively disclose certain personal details about the user. As mentioned before, IRMA guarantees several security properties like verifiers that can validate the source and integrity of the attributes. However, there is as of yet no security property that can protect the user against *shoulder surfing*. Shoulder surfing is a type of social engineering where a (malicious) person looks over the shoulder of the user. In our case, that would be the user of the IRMA mobile app. While the QR code is shown on a web page to 'load' attributes from an issuer, there is a possibility that a malicious user could scan this QR with his own mobile phone before the intended user scans it. The requested attributes would then be loaded within the mobile app of the malicious user. The malicious user could use these attributes to sign digital content without actually owning the attributes.

This ‘limitation’ is comparable to someone watching over your shoulder while typing in your password and hitting enter before you do. The problem is a much bigger societal issue and not exclusively related to IRMA. However, it is important to note that some of the responsibilities still lie by the user when protecting his own privacy. Within the IRMA project they are currently working on a solution to prevent problems like shoulder surfing by using a so called ‘pairing code’.

7.6.3 Identity Falsification

IRMA cannot always guarantee the prevention of *identity falsification.* Before a user can complete one of the IRMA sessions, a PIN code is required. This prevents IRMA sessions from being completed when the PIN code is unknown to the user, e.g. when the mobile phone is stolen. However, there is still a possibility of identity falsification. An example would be a user, who is below the age of 18, that wants to buy a concert ticket. The concert is only for people over the age of 18. The website, where the ticket is ordered, only requests the address and age of the person who is buying. The address attribute is used to deliver the ticket, while the age attribute is used to verify if the user is old enough to own a ticket. Since the user cannot prove that he is over the age of 18 with his own IRMA mobile app, he uses the app of his older sibling. There is no way for the ticket seller to verify if the ticket is sent to the right person. The service provider should therefore be careful on which attributes are requested. On the other side, the advantage of IRMA here is that the service provider has the possibility to do so. It can request more personal data about the buyer, if there is a need, while still conforming to the legal obligation of data minimization.

Examples like shoulder surfing and identity falsification are societal issues that are not easily solved by any system or toolset, including IRMA. Although stating that users still have certain responsibilities regarding the protection of their own privacy, we aim to help the user as much as possible to protect their privacy. Therefore, we will combine the theoretical and practical knowledge gathered so far and translate the finding into an attribute-based signature toolset with IRMA as its basis. The toolset aims to find the right balance between disclosing relevant personal information of the publisher to the reader, while not disclosing an excessive amount of personal information of the publisher when signing digital content. Giving the publisher more control over their privacy while signing digital content, while giving the reader the ability to easily verify the authenticity of the digital content.

8 Toolset Implementations

> Q5: *"What kind of application(s) should be designed and developed as part of the toolset to realize accessible and comprehensible digital signatures?"*

Now that we have established what the possibilities (and limitations) of IRMA are from a theoretical perspective, we want to look at what is actually possible in the practical sense. As discussed earlier in chapter 2 'Methods', we can strengthen the fundamental research by performing applied research. The applied research consisted of figuring out what is necessary to develop a toolset that can (1) *attach* an attribute-based signature to any type of digital content, (2) *verify* the signature when attached to some digital content, and (3) make it *accessible* to everyone.

The decision was made that the toolset should consist of two 'applications'. The first application being the *IRMA Signature Application* or ISA. The ISA is based on an older project[32] develop by the Privacy by Design Foundation. The goal of this older project was to develop an application that can create IRMA signature requests and send them to an e-mail client. The ISA expands on this application and makes it possible for users to select any type of file (digital content) and attach an IRMA attribute-based signature to it. Removing the ability to send IRMA signature requests to an e-mail client. Additionally, the ISA gives the user the ability to verify the signature when attached to an attribute-based signed file. The second application is a browser plugin named IRMA Signature Plugin (ISP). This second application is described as a proof of concept and demonstrates a potential idea to realize the browser plugin. The third requirement of the toolset is to make it accessible to everyone, therefore the use of a browser is a logical choice. Everyone who accesses the internet makes use of a browser, plus the installation of a plugin is not a difficult task in general. The two applications combined form the toolset and satisfy the set requirements. The requirements for the ISA are discussed in this chapter, including which techniques are necessary to develop the complete attribute-based signature toolset. To prioritize the requirements, the MoSCoW method is used.

8.1 Signature Type

Before ISA is developed it should be clear how the signature is going to be attached to the digital content. Note that the digital content can be any type of file like a (text) document, image or video. There are three distinct possibilities to 'store' a signature: *Detached*, *Enveloped*, and *Enveloping*:

- **Detached Signature**: The signature is stored in a separate file. The format can be freely chosen.
- **Enveloped Signature**: The signature is embedded within the original digital content. The format of the original digital content has to support this. Examples of such formats that allow embedding signatures are PDF and XML.
- **Enveloping Signatures**: A new file is created within an established signature format and used that as a container, embedding the original digital content. The most common format is the Cryptographic Message Syntax [Hou99] .

[32] https://github.com/privacybydesign/irma$_{s}ignature_{a}pp$

Since we have to cope with the limitation of IRMA only allowing to sign a message (string) using an attribute-based signature, the decision was made to use the hash-and-sign scheme, as mentioned earlier in chapter 7.6.1 'Hash-and-Sign Scheme'. The hashed base64 encoding of the digital content is used as the input of the message entity within the IRMA attribute-based signature. This results in an enveloping signature, where the digital content is embedded into the signature as a string. Followed by the attribute-based signature being embedded within a newly signed file that contains, among other information, the original file (base64 encoded) and the attribute-based signature.

8.2 IRMA Signature Application (ISA)

The IRMA Signature Application (ISA) is a desktop application and functions as a service provider within the IRMA system. ISA gives the user the ability to sign any type of file (digital content) using IRMA's attribute-based signatures and verify any type of file that is signed with an attribute-based signature. The aim of this thesis is to make digital signature semantics comprehensible. Therefore, the user should be able to sign and verify signatures that display comprehensible semantics. Since this thesis is performed within the field of computing science, the focus will be on the technical functionalities instead of the 'soft' functionalities like user experience. Even though 'soft' functionalities are important in making the toolset accessible and easy to use, it is out of scope for this thesis, as mentioned before in the chapter 2.3 'Scope'.

8.2.1 Core Processes & Functionalities

The core processes of ISA are split up between two parts, SIGN and VERIFY. Where SIGN relates to the functionalities that are required for the *signing* process to work. And where VERIFY relates to the functionalities that are required for the *verify* functionalities to work.

The **SIGN** process consists of the following steps:

1. Select a file to sign (local).
2. Encode the file to a base64 variant.
3. Hash the Base64 encoding.
4. Relevant attributes are selected as part of the signing 'policy'.
5. An attribute-based signature session request is made.
6. The signature request is confirmed by scanning the QR and filling in the PIN code using the mobile app.
7. Selected attributes are appended to the signature, as verified claim for the policy.
8. ISA receives the attribute-based signature.
9. ISA attaches the received signature to the file.
10. The signed file can be saved (local).

When a user wants to *sign* digital content, let's take a PDF file, he can select this locally from his computer. In the background, the selected file gets encoded to a base64 encoding. This encoding is necessary because we want to load the file into the message entity of the attribute-based signature. And since this can only be in the form of a string, we need to convert the file (bytes) to a string, using base64 encoding. Once the file is converted to a base64 encoding, the encoding is then hashed using the secure SHA-3 hashing algorithm. This results in the string being a fixed number, with a reasonable string size to be loaded into the message entity of the signature.

The user must give the signature a 'name'. This name will be used as the name of the resulting signed file. The user can select the attributes that will be assigned to the signature as part of the 'policy'. The policy is introduced because not attaching any attribute when signing would defeat the goal of making the signature semantics comprehensible. A signature request can now be initiated. For the signature request to succeed, the user needs to scan the QR code and fill in his PIN code using the IRMA mobile app. If successful, the selected attributes are attached to the signature and returned to ISA. The user can now decide to export the signed file to his local computer. The signed file now consists of the base64 encoding of the file and the related IRMA attribute-based signature. The complete SIGN process is visualized in figure 6.

SIGN Process

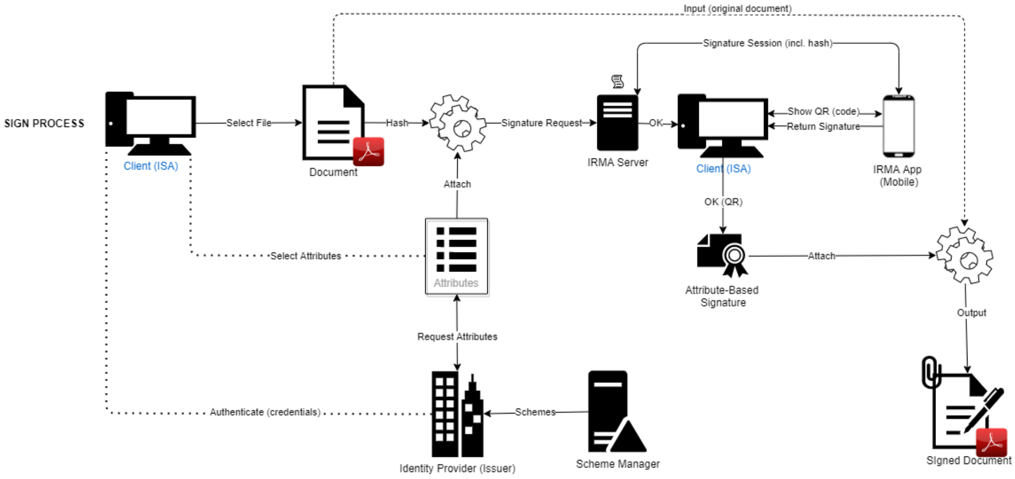


Figure 6: SIGN Process.

The **VERIFY** process can be performed by the IRMA signature application and IRMA Signature Plugin. Both applications act as the requestor. VERIFY consists of the following steps:

1. Select a signed file (local).
2. Request to validate the signature embedded within the signed file.
3. The requestor receives the signature status.
4. The requestor displays the signature status, related attributes, and metadata of the file.

When a user wants to *verify* a signed file, he can select this locally from his computer. Both the original file (e.g. a PDF) and the IRMA attribute-based signature are extracted from the signed file. The attribute-based signature is processed and validated. When the attribute-based signature is validated, the message field will be extracted, containing the hash of the original file. The original file is also stored in the signed file as a base64 encoding. To check if there were no changes to the original file, the base64 string of the original file is hashed. The resulting hash is then compared with the hash from the message field of the attribute-based signature. If the signature is valid and the two hashes are equal, the verification (VERIFY) process is successful. The signature status and other related information, like the attached attributes and original document metadata, are displayed by the requestor. The complete VERIFY process is visualized in figure 7.

VERIFY Process

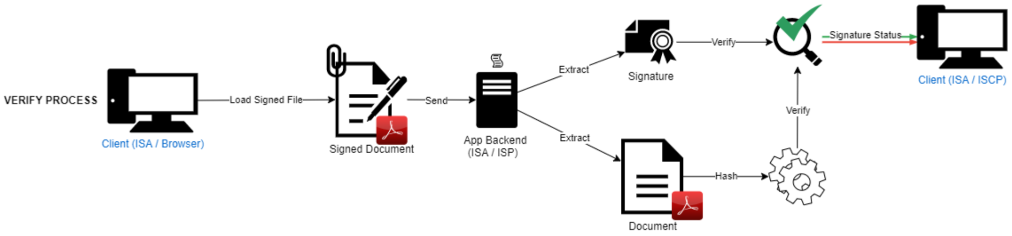
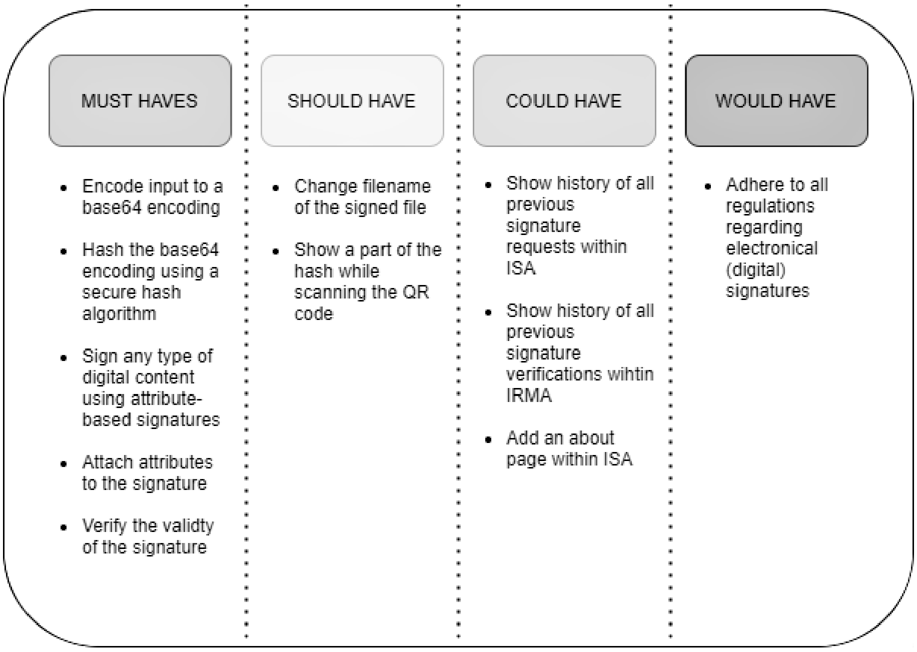


Figure 7: VERIFY Process.

8.2.2 Requirements

Q6: *"Which requirements must the toolset satisfy before it can be deployed (to the public) as a minimal viable product?"*

As mentioned before in chapter 2 'Methods', during the development phase and part of the applied research, the MoSCoW requirement prioritization method is used to scope the toolset and prioritize the functionalities. In figure 8 the requirements of the toolset are displayed in a simplified manner using the MoSCoW method.



MUST HAVES	SHOULD HAVE	COULD HAVE	WOULD HAVE
• Encode input to a base64 encoding • Hash the base64 encoding using a secure hash algorithm • Sign any type of digital content using attribute-based signatures • Attach attributes to the signature • Verify the validty of the signature	• Change filename of the signed file • Show a part of the hash while scanning the QR code	• Show history of all previous signature requests within ISA • Show history of all previous signature verifications wihtin IRMA • Add an about page within ISA	• Adhere to all regulations regarding electronical (digital) signatures

Figure 8: Toolset Requirements.

The must haves are essential for the application to function as expected. The implementation of the 'hash-and-sign scheme' can be seen in the first three must have requirements. These requirements define the ability of the user to sign any type of digital content. Followed by the actual ability to sign and verify the digital content and signature. The related 'sign-screen' and 'verify-screen' can be seen in appendix A. Changing the name of the resulting (signed) file and displaying the hash of the encoded file on the QR-screen are both not essential to the functioning of the application. Instead of choosing a name, the name of the selected file could be taken. Showing the hash could be left out. However, this would result in the user not being able to validate the hashes. Finally, an about page was added to provide the user some additional information about the application. All other requirements were not met, but as mentioned before are not essential to the functioning of the application and therefore not part of the minimal viable product.

8.2.3 Results

In appendix A all screens are displayed that are part of the current version of the ISA, including additional details about the functionalities that each screen provides to the user. The IRMA Signature Application (ISA) in its current state provides the user the following main functionalities:

- *Sign* any type of digital content with an attribute-based signature. The user can select any available attribute in his IRMA mobile app and include the attribute in the signature. Information is displayed about the selected digital content in the form of metadata and the hash. The user can validate the hashes by comparing the hash as part of the message entity of the signature (shown when confirming the signature request in the IRMA mobile app), and the hash is shown when scanning the QR code. If equal, the user can be sure that he is signing the correct digital content.
- *Verify* any type of digital content that is signed with an attribute-based signature using the ISA. The proofstatus (validity) of the signature, the hash (as part of the message), and the attached attributes are shown to the user. This gives the user an overview of who signed the content and the authenticity of the source. Additionally, the user can again check whether the hash is still correct since the hash is also displayed when verifying the signature using the ISA.

The final result of the IRMA Signature Application allows the user to sign any type of digital content using attribute-based signatures. Verifying the signed content gives the user a comprehensible overview of what attributes are attached to the signature. The attributes themselves provide the user with an overview of the identity of the signer. Therefore, the application satisfies the goal of this thesis to make digital signature semantics easier to comprehend.

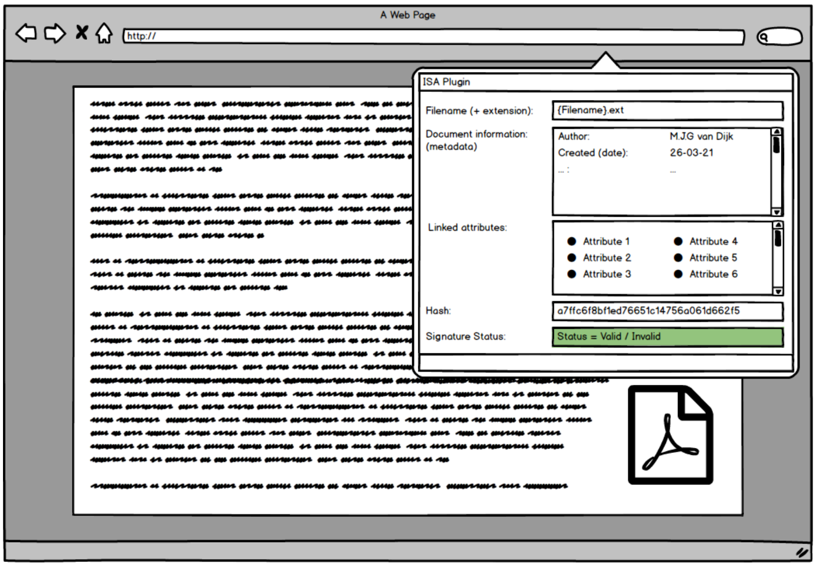
8.3 IRMA Signature Plugin (ISP)

The idea of using a browser plugin was introduced because of the accessibility advantages and the potential help it can deliver when battling against disinformation that is received via the web. Plugins are used as an 'extension' to the browser. Most people are familiar with using browsers and integrated plugins. For the toolset to become part of day-to-day usage by the 'average' internet user, it should be easily accessible. Apart from accessibility, the plugin can also provide the reader (verifier) with information about the attribute-based signed file received via the web. Plugins are cross-platform since they are 'integrated' into browsers. The goal of the thesis is to increase the comprehensibility of digital signatures semantics, and thereby high informational value of the information displayed by the signature is crucial. Additionally, a browser plugin can realize the sub-goal of increasing the level of accessibility for readers (verifiers) when verifying an attribute-based signed file via the web. Note that it is impossible to use the plugin to sign digital content using IRMA attribute-based signatures; only the signature verification is part of the proof-of-concept. In theory, it should be possible to integrate the sign functionalities into the plugin, though it would need more research on how this would practically be possible to achieve.

8.3.1 Design (sketch)

The decision was made to design a proof-of-concept sketch. The proof-of-concept gives an idea of what can (technically) be possible when integrating the signature verification process of IRMA into a plugin. The sketch shows how the plugin could be designed and what functionalities are required before it can be considered a minimal viable product. A sketch of the plugin's minimal viable product is shown in figure 9. The minimal viable product would consist of the following functionalities: the filename including the extension type of the original document, document information in the form of the available metadata, the attributes that are part of the attached attribute-based signature, a hash of the original file, and the proof status of the signature that can either be valid or invalid. The file should be recognized by the plugin, either by a URL that links to a document on the page, or a document loaded within the browser. The document can be both loaded via the web or locally within the browser. After the plugin recognizes the document, the user has the option to verify the document.

Looking at the information that can be extracted from documents like a PDF, in combination with the information available within the attribute-based signatures, the IRMA Signature Plugin should at least consist of the following information: *Filename*, *Document Information*, *Attributes*, and *Signature Status*. The filename is, as the name suggests, the name of the attribute-based signed file. The 'Filename' is saved in the following form: Filename.ext.irmasignature. Where 'ext' is the extension name of the original file (digital content). The 'Document Information' part displays metadata related to the original document. This can include the name of the document, creation date, author, and many more. This informs the reader about the original document. This field is filled depending on the metadata available within the original document. The linked 'Attributes' are extracted from the attribute-based signature. In the SIGN process, several chosen attributes are attached to the signature. These will be displayed in this list. The hash, part of the message entity of the signature, is shown for the user to compare to the hash of the original file. Note that the user needs an alternative way to get the hash of the file loaded within the browser. In the background, the plugin can automatically hash the file that is loaded within the browser, followed by comparing the resulting hash to the hash available in the signature. If the hashes match, it means that the document has not changed since signing (guaranteeing integrity). Finally, the ‘Signature Status’ is shown. This displays whether or not the signature is valid and if the document has not been altered (matching hashes). In figure 9 a design sketch is made that gives an idea of how the plugin could be realized.



Filename (+ extension): {Filename}.ext

Document information: (metadata)

Author: M.J.G van Dijk

Created (date): 26-03-21

... : ...

Linked attributes:

- Attribute 1
- Attribute 2
- Attribute 3
- Attribute 4
- Attribute 5
- Attribute 6

Hash: a7ffc6f8bf1ed76651c14756a061d662f5

Signature Status: Status = Valid / Invalid

Figure 9: IRMA Signature Plugin Sketch.

8.3.2 Differences with ISA

Compared to ISA, the plugin (ISP) is not very different in regards to the signature verification process. Both the applications allow the user to verify an IRMA attribute-based signature and show the related metadata, attached attributes, hash, and proof status. The functional differences with ISA are present in two ways:

1. *Accessibility*: the plugin has a greater level of accessibility to the user. The plugin can be used cross-platform and allows users to easily verify the validity of a signature that is received via the web.
2. *Signage*: the ISA allows users to sign digital content, while this is not possible in this (proof-of-concept) plugin. The plugin is designed to verify signature exclusively. For now, the ISA must be used to disclose attributes, attach them to a signature, and sign the content with this signature.

8.3.3 Future Work

Potential future work could consist of realizing the plugin on the basis of the sketch shown in figure 9. This can be realized in any browser that supports the creation of plugins. The plugin recognizes when a file is loaded within the browser. The file can then be extracted, encoded (base64), and hashed (SHA-3). The resulting hash will be compared against the hash within the message entity of the attribute-based signature, followed by validating the signature in total. Another option could be to scan a web page for URLs that links to a document. The document could be downloaded and checked if signed with an IRMA attribute-based signature. If this is the case, the same VERIFY process can be performed, resulting in a valid or invalid signature status. If the IRMA Signature Plugin could be realized, it has the potential to be a suitable solution to increase the accessibility of the toolset, allowing it to run on all platforms (desktop, laptop, tables, mobile phones etc.). When the accessibility of the toolset increases (by developing a plugin available cross-platform), the chance of more people using the toolset becomes higher. Ultimately, leading to mainstream adoption.

Another, more advanced, possible implementation would be a native integration of the plugin into a browser, like we see with TLS (or SSL). TLS has made its way into the native functionalities of a browser. Something that is also possible with (attribute-based) signatures. Where, the plugin and its functionalities are embedded within a browser. While users of a browser surf the web, the browser automatically recognizes whether the loaded digital content (on a web page) is signed or not. Followed by somehow displaying to the user what the proof status of the signature is. Additionally, the browser can give the user the ability to find out more about the signature. Again, an attribute-based signature would give the user more (useful) information about the signature and in particular the signer's identity in comparison to a 'standard' signature.

8.4 Final Toolset Result

The plugin makes verifying an IRMA attribute-based signature more accessible to the public, in comparison to the ISA. The plugin has the same functionalities as the ISA when it comes to verifying the signature. The plugin does not allow the user to sign digital content. Instead, the ISA or any other application that can sign digital content using IRMA attribute-based signatures should be used. The proof-of-concept design sketch has shown how the plugin (ISP) can be realized in terms of design and (functional) requirements. Overall, higher accessibility of verifying IRMA attribute-based signatures would benefit the adaption of IRMA as a privacy-friendly system.

Additionally, the ISA gives the user even more functionalities, although the application is less accessible than the plugin. ISA has shown (see appendix A) that it is possible to develop an application that can give the user both the ability to sign digital content using attributes and verify the signed digital content. Combining both the power of the ISA and the ISP, the user has access to an accessible process of signing and verification. At the same time, the semantics of the signature are both comprehensible and informative.

8.4.1 Minimal Viable Product

The minimal viable product as required within this thesis is made out of at least one application that allows the user to sign digital content using IRMA attribute-based signatures and verify this signed digital content while showing comprehensible and informative information about the signer to the verifier. This also means that the signer should have the possibility to attach informative information, like personal details, to the signature. The 'must haves' at figure 8 show in a structured manner what the requirements of the minimal viable product are in technical terms. In our case, the ISA meets all the must have requirements, including all should and one of the could have requirements. As stated within the requirements figure, the realization of the plugin (ISP) is something for the future and is therefore not part of the minimal viable product. The ISA already allows users to verify digital content signed with an attribute-based signature.

8.4.2 Comprehensibility and Informational Value Analysis

Digital signatures should be comprehensible and provide an higher level of informational value. Whether this increase in comprehensibility and level of informational value provided by the toolset is indeed 'sufficient' is out of scope for this thesis and should be researched further. For now it at least provides additional (identifying) information about the signer to the verifier, while presented in a more comprehensible manner in comparison to a 'standard' signature. This is theoretically exemplified in chapter 4.3.1 'Informational Value', and practically exemplified in this chapter and appendix A.

The decision was made to use the attribute-based signature as a variant of signature that can provide an increase in informational value, while simultaneously making the signature more comprehensible. The technique of ABCs allows users to selectively disclose attributes. These attribute can then be assigned to a digital signature, and since attributes provide (personal) information about the signer it increases the informational value of the specific signature. On the other hand, the user that verifies the digital content, signed with an attribute-based signature, has more valuable information (the attributes) to validate. And since the attributes are in a human-readable format, it is not difficult for the verifier to comprehend what the attributes mean. Additionally, the attributes are given out by certain issuers, like municipalities or universities. In short, organizations that are allowed to give out (identifying) information, like a diploma or driver's license. Therefore the verifier can be certain that the assigned attributes are valid. The toolset combines these theoretical possibilities and enables the users to sign and verify attribute-based signatures, utilizing the functionalities provided by the technique of ABCs and IRMA.

9 Discussion

In this chapter, several use-cases are discussed. The use cases give an idea on how the developed toolset can be used in-practice and what attribute-based signatures can deliver in general. Potential further research regarding both the toolset and IRMA is also discussed. Finally, several alternative projects are listed. All of these alternative projects are developing attribute-based technologies in some way. Therefore, a comparison is made between these alternative projects and IRMA.

9.1 Use-cases

Attribute-based signatures give people a selective identity when signing. Applications, like those in the toolset mentioned in chapter 8 'Toolset Implementations', give people the tools to do so. The reasons to use an attribute-based signature can vary. E.g. governments signing digital content to become more trustful towards their citizens, or individuals like an artist signing his digital work to prove his authorship. The goal to (selectively) disclose certain attributes can also vary. By giving a few example use-cases, we can discover what the potential is of an attribute-based signature backed by an easy-to-use application like ISA.

Logical use-cases are e.g. to sign a document with a certain identity, like a doctor signing an official medical document. Governments signing an official video statement to prevent deep-fakes from being confused with the official one. Webshops only requiring the address and payment details of the user, which makes the webshops automatically adhere to the data-minimization requirement as stated within the GDPR. You can list a great number of similar use-cases, all focusing on giving a selective identity to the signed content. In the following chapters, we will expose some 'unordinary' use-cases, revealing the rich potential of attribute-based signatures.

9.1.1 Use-Case #1: Freedom of Information

Worldwide there are many governments that hold themselves to some sort of *Freedom of Information Act.* This act ensures that governments are transparent in sharing their 'known' information, or knowledge. When this act is 'requested', all involved parties want to be sure that no false identities are used. Governments, who are in this case always the 'publisher', want to ensure that the information they reveal can always be linked back to them and not to another (malicious) party claiming they published the information. It would harm the reputation of the government if alternative information were published by a malicious party, e.g. rivaling governments or activists. Therefore, it is essential for the publishing government to sign the information. Or even better, sign the information with identifiable attributes. This allows the receiver to easily understand the information on the signature and determine if the received act is truly coming from the original source, the government. Both parties benefit from a transparent and comprehensible signer identity on both the signature and the original source. In this use-case of a freedom of information act request, the need for such transparency becomes even more clear.

This use-case can also be approached from a ‘warning’ perspective. Governments often have a certain digital communication tool that distributes warnings to their citizens in the event of something dangerous or even life-threatening, like a tornado, flooding, or toxic gas leak. In all of these events, the government wants to inform its citizens with speed and clear authenticity. There is no time for the citizens to doubt the authenticity of the source. The identity of the author, in this case the government, should be clear immediately. Signing the message that includes the warning with attribute-based signatures would give the reader an immediate and clear picture of the identity of the signer. Strongly reducing the time it takes for the reader to validate the authenticity of the message (warning). This is especially relevant when time is of the essence.

9.1.2 Use-Case #2: Pseudonym

Another possibility is to sign content with a *pseudonym*. The attribute can be of a certain type that only reveals a pseudonym of the identity of the user. This does not provide a full identity of the signer, but can result in certain trust. Imagine a writer (publisher) always publishing and signing articles with an attribute that does not disclose his full identity, meaning that the content is signed with a pseudonym attribute. The reader (verifier) cannot determine what the identity of the writer is, but can note the pseudonym attribute. The next time the write published an article signed with the same pseudonym attribute, the reader recognizes this same pseudonym attribute. Even though the writer does not provide a full identity, he can build trust by the readers. Especially to (citizen) journalists within war zones or nations with poor freedom of speech this can be a relevant solution since only a pseudonym of the journalist’s identity is revealed. The reader judges the content and shapes a level of trust related to the writer. Building trust without revealing identity.

9.1.3 Use-Case #3: Source Protection

Source protection, sometimes referred to as source confidentiality or reporter’s privilege, is a right accord under international law. Source protection states that authorities, including courts, are prohibited from forcing a journalist to reveal their identity when they choose to write anonymously. This came forth out of the idea that journalists would become reticent to share information of public interest when the anonymity of the journalist cannot be guaranteed. Although source protection states that the privacy of the journalist should be guaranteed, in practice this risk is still present. Laws could be ignored by rogue authorities and demand journalists to reveal their identity. To mitigate this risk, journalists have the option to sign their articles with a digital signature, in particular attribute-based signature to provide selective disclosure of their identity. This could e.g. be a pseudonym attribute like mentioned before, preventing the journalist from revealing their full identity but disclosing enough information for the reader to determine if the content can be trusted.

We could also look at this from a different perspective. A publisher of newspapers perhaps wants to provide source protection for their employees. When a certain, potentially sensitive, article is published the writer (journalist) could be holding back on signing the article using his identity. Alternatively, the article can be re-signed by the publisher as an organization. In this case, the publisher (employer) takes the responsibility and provides its employees with source protection.

9.1.4 Use-Case #4: Fraudulent Resumes

Lying about your resume is not an uncommon matter. Surveys that tried to reveal the percentage of people lying on their resumes ranged from around 25 to 30%[33]. This is a significant amount of people willing to lie about their skills when creating their resume. For employers, it can be very annoying when a significant amount of the received resumes do not correspond with the skillset of the employees.

By creating the resume using authenticated attributes, the employer can be certain that the employee indeed has the submitted skills present within the resume. Additionally, the final resume can be signed with the same attributes as present in the resume. Guaranteeing both the integrity and authenticity of the final resume. Preventing employees from lying about their skills and helping employers saving time when filtering resumes.

9.2 Toolset In-Practice

The different use-cases have shown that an attribute-based signature toolset can be of use in multiple scenarios. Specifically related to the toolset discussed in this thesis, the IRMA Signature Application (ISA) is mainly intended for users who want to sign certain digital content. However, ISA also allows the user to validate the signature. This can be useful when someone just signed some digital content and wants to validate if the signature is indeed correct. Using ISA to display that the correct content is signed and that the attached attributes correspond with the attributes provided when signing. Technically ISA can also be used by a user who wants to exclusively validate a signature. The user can validate if the signed content is valid, and check which attributes are related to the signature. These functionalities combined realize the goal to make digital signatures semantic more comprehensible and available to the public.

Regarding the accessibility of (attribute-based) signatures, the (proof-of-concept) browser plugin is a better contender. ISA is a desktop application and therefore not as easy in day-to-day use for the average user in comparison to a browser plugin, especially when browsing the internet is the main activity of most users. Browsers are not only used on desktop computers but also on tablets and mobile phones. Therefore a browser plugin was the logical choice, allowing people to validate signatures within their browser and additionally making the application cross-platform. The browser plugin proof-of-concept showed that realizing an application that can directly validate attribute-based signatures within the browser has a benefit to the accessibility of the toolset. In terms of comprehensibility of the signature semantics, the ISP is comparable to the ISA.

Combining both the ISA and ISP, the toolset allows for an improvement in both the comprehensibility of digital signature semantics and the accessibility in using attribute-based signatures.

9.3 Further Research

Further research regarding the toolset should mainly focus on improving the user experience of the IRMA Signature Application and the realization of the IRMA Signature Plugin proof-of-concept. Although the ISA is a functioning application, it would still

[33] https://www.monster.com/career-advice/article/the-truth-about-resume-lies-hot-jobs

require more testing before it could be deployed to the public. So, both the ISA and the ISP are an example of how attribute-based signatures can be realized in practice but are currently not in a state where they can be used by the general public. Further research and development have to be performed in order to make the toolset reasonable in use and available to the public.

Regarding IRMA, the further research should focus on allowing more types of digital content to be signed using an IRMA attribute-based signature instead of only allowing a single message to be signed. It would open up new possibilities for IRMA to be an even better front-runner of practical attribute-based signage. Embedding IRMA attribute-based signature into existing platforms, like text editors or browsers. Instead of using workarounds like the hash-and-sign scheme, the addition of multiple types of digital content signing would make the integration of IRMA attribute-based signatures into other applications easier and could increase the level of usability in general. Ultimately, we want to see the integration of attribute-based signatures into the daily life of the 'average' internet user, making it harder for disinformation to have a negative impact on society.

9.3.1 Alternative Projects

There are several projects other than IRMA that try to achieve similar goals using attribute-based technologies. As an extension to the potential further research, we will briefly discuss some of the projects and compare them to IRMA.

- Decode[34]: *Decode* describes itself as "an experimental project to develop practical alternatives to how we use the internet today". DECODE provides tools that put individuals in control of whether they keep their personal information private or share it for the public good.
- Schluss[35]: *Schluss* claims to return the control of information back to the user. The technology behind Schluss is also open source.
- Serto[36]: *Serto* claims to allow users to utilize decentralized technology to make data more portable, private, and valuable. A platform is developed that allows users to know the source of data and verify the issuer of the data. Turning 'free-form' data into verifiable credentials and making them issueable to others.
- Sovrin[37]: *Sovrin* is a non-profit foundation, comparible to the Privacy by Design foundation, and claims personal management of digital IDs using the Sovrin Network. The Sovrin Network is an open source project creating a public utility for self-sovereign identity.
- SelfKey[38]: *SelfKey* is a blockchain startup developing digital identity solutions. One of these solutions is the SelfKey wallet. This wallet should give users full control of their data, documents and digital assets.

[34] https://decodeproject.eu/
[35] https://schluss.org/nl/
[36] https://www.serto.id/
[37] https://sovrin.org/
[38] https://selfkey.org/

Attribute-Based Projects					
	Authentication	Signatures	Open Source	In-Practice	Decentralization
IRMA	Y	Y	Y	Y	Y
DECODE	Y	N	Y	Y	N
Schluss	Y	N	Y	Y	N
Serto	Y	N	Y	N	Y
Sovrin	Y	N	Y	Y	Y
SelfKey	Y	N	Y	Y	Y

Table 1: Attribute-Based Projects Table

A common trend among these alternative projects is that they use attributes and credentials for users to issue and disclose (or sometimes referred to as 'sharing') when authenticating to a service provider, often comparable to the way IRMA implements these. All provide some way to let users have more control over their data, whether personal or not. IRMA stands out in the possibility to use the attributes to sign digital content. All of the other projects seem to purely focus on giving users a 'safe space' to store credentials and disclose or share them to service providers when they see fit.

The other interesting difference between the projects is the use of decentralization techniques. IRMA, Serto, Sovrin, and SelfKey all use some form of decentralization. However, there is a difference between the way decentralization is implemented. Serto, Sovrin, and SelfKey all make use of blockchain technology. SelfKey claims that "by using blockchain technology, *Self-Sovereign Identity* puts users back in control of their personal data". However, IRMA has a different approach to decentralization. As we have seen before in chapter 5.3.2 'Tracking and Decentralization' when using IRMA, no third party is involved when the user is disclosing attributes or singing attribute-based signatures. This meant that regarding the eIDAS regulations, it only satisfies the level of 'advanced' digital signatures. However, it does allow for a decentralized setup where no third party is involved. A 'peer-to-peer' structure is created where the user discloses the attributes directly to the service provider.

All in all, there seem to be more projects of interest that try to utilize attribute-based technologies, whether it relates to authentication or signage. Though, we see that all projects except IRMA focus more on the notion of 'self-sovereign identity' and do not include any solution to utilize the attribute-based technology to sign digital content. Regarding further research, it would be interesting to look into the potential of using the blockchain. Blockchain can support the development of a decentralized digital identity(-management) system. After that more research can be performed on which of these systems is 'better'. Either a decentralized identity system using blockchain, a partial decentralization like in IRMA, or a centralized identity system. The projects mentioned above can be used as a starting point in comparing several existing digital identity systems and ideally use the results to improve the existing technique behind attribute-based signatures.

10 Conclusion

MQ: *"What kind of toolset should be developed to increase the comprehensibility of digital signatures semantics, in particular to support the battle against disinformation?"*

Within this thesis, research was performed from a ‘battle against disinformation’ perspective. Consequently, it came forward that current countermeasures often neglect the importance of source authenticity. As an alternative countermeasure, the use of digital signatures was researched. It came forward that the current implementations of ‘basic’ digital signatures lacked comprehensible semantics. To counter this lack of comprehensible semantics, a toolset was developed. Ultimately, supporting the battle against disinformation.

The first part of the main research question can best be answered by looking back at chapter 8 ‘Toolset Implementations’. This chapter describes what kind of toolset can be developed to increase the comprehensibility of digital signatures. Resulting in a toolset consisting of both the IRMA Signature Application (ISA) and the IRMA Signature Plugin (ISP). The ISA has a key role in providing the answer to the main question. This application shows how users can sign digital content with IRMA’s attribute-based signatures and verify digital content that is signed with these signatures. Showing that when users can verify an attribute-based signature, the semantics of that signature is less difficult to comprehend. Additionally, the signer has the flexibility to sign any type of digital content with any attribute he or she possesses. Gaining more control over their privacy, in comparison to ‘basic’ digital signatures.

The signing of digital content in the current version of IRMA is not very generic. In the current version, only strings can be signed with attribute-based signatures. To resolve this limitation this thesis introduced the hash-and-sign scheme. Making it possible for the users of the toolset to sign any type of digital content.

To increase the accessibility of the toolset, the IRMA Signature Plugin (ISP) was introduced. The plugin should make it possible for the user to verify digital content signed with attribute-based signatures on multiple platforms. For the toolset to adopt into the day-to-day life of most (internet) users it should be more accessible. The proof-of-concept of the plugin shows a potential realization. The plugin is a promising concept that should be researched further, making the toolset more accessible and increase the chance of mainstream adoption.

The second part of the main research question can best be answered by looking at the use-cases. Various use-cases show that the toolset can provide support in the battle against disinformation. Whether it be a journalist trying to spread news directly from a war zone, or a government trying to inform its citizens. The use of the toolset makes verifying signed content comprehensible to all, while the authenticity of the source is guaranteed.

So, instead of using countermeasures against disinformation that restrict people in forming their own thought. This thesis presents a tool that gives people the ability to sign and verify attribute-based signatures, making the semantics of a digital signature easy to comprehend. Consequently, it is easier for people to define the authenticity of a source, making it harder for disinformation to have a negative impact on society.

References

[AJ13] Gergely Alpár and BPF Jacobs. Credential design in attribute-based identity management. 2013.

[BCD+14] Patrik Bichsel, Jan Camenisch, Maria Dubovitskaya, R Enderlein, Stephan Krenn, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Janus Dam Nielsen, Christian Paquin, et al. D2. 2 architecture for attribute-based credential technologies-final version. *ABC4TRUST project deliverable. Available online at https://abc4trust. eu/index. php/pub*, 2014.

[BCD+17] Foteini Baldimtsi, Jan Camenisch, Maria Dubovitskaya, Anna Lysyanskaya, Leonid Reyzin, Kai Samelin, and Sophia Yakoubov. Accumulators with applications to anonymity-preserving revocation. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 301–315. IEEE, 2017.

[Bra00] Stefan Brands. *Rethinking public key infrastructures and digital certificates: building in privacy.* Mit Press, 2000.

[Cav09] Ann Cavoukian. Privacy by design. 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *International conference on the theory and applications of cryptographic techniques*, pages 93–118. Springer, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *International Conference on Security in Communication Networks*, pages 268–289. Springer, 2002.

[Cou14] Council of European Union. Council regulation (EU) no 910/2014, 2014. `https://eur-lex.europa.eu/eli/reg/2014/910/oj`.

[Cou16] Council of European Union. Council regulation (EU) no 2016/679, 2016. `http://data.europa.eu/eli/reg/2016/679/oj`.

[Cul15] Nicholas John Cull. *Counter propaganda: Cases from US public diplomacy and beyond.* Legatum Institute, 2015.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[Fal15] Don Fallis. What is disinformation? *Library trends*, 63(3):401–426, 2015.

[GBN18] Richard Gunther, Paul A Beck, and Erik C Nisbet. Fake news may have contributed to trump's 2016 victory. *Unpublished manuscript. Retrieved from https://www. documentcloud. org/documents/4429952-Fake-News-May-Have-Contributed-to-Trump-s-2016. html*, 2018.

[GC16] Lucas Graves and Federica Cherubini. The rise of fact-checking sites in europe. 2016.

[GG12] Lucas Graves and Tom Glaisyer. The fact-checking universe in spring 2012. *New America*, 2012.

[HAvdBJ15] Brinda Hampiholi, Gergely Alpár, Fabian van den Broek, and Bart Jacobs. Towards practical attribute-based signatures. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 310–328. Springer, 2015.

[Hou99] Russell Housley. Cryptographic message syntax. Technical report, RFC 2630, June, 1999.

[IBM12] IBM Research Zürich Security Team. Specification of the identity mixer cryptographic library, version 2.3.4., 2012. Technical report, IBM Research.

[JMV01] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1(1):36–63, 2001.

[JS20] Bart Jacobs and Hanna Schraffenberger. Friction for privacy. why privacy by design needs user experience design. 2020.

[MPR11] Hemanta K Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In *Cryptographers' track at the RSA conference*, pages 376–392. Springer, 2011.

[PGC$^+$20] Ivan Petrov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Jian Jiang, Luis RP, Sheng Zhang, Pingyu Wu, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[SY08] Guo Shanqing and Zeng Yingpei. Attribute-based signature scheme. In *2008 International Conference on Information Security and Assurance (ISA 2008)*, pages 509–511. IEEE, 2008.

[UB13] Joseph E Uscinski and Ryden W Butler. The epistemology of fact checking. *Critical Review*, 25(2):162–180, 2013.

[VA13] Pim Vullers and Gergely Alpár. Efficient selective disclosure on smart cards using idemix. In *IFIP Working Conference on Policies and Research in Identity Management*, pages 53–67. Springer, 2013.

A Appendix: ISA Screens

The following chapters will display the IRMA Signature Application screens present within the current version of the application. The chapters are divided between *Home-screen*, *Sign-screen*, *QRscan-screen*, and the *Verify-screen*. For convenience, the *About-screen* is excluded from the appendices. The about page describes the application but does not provide any relevant information to this thesis.

A.1 Home

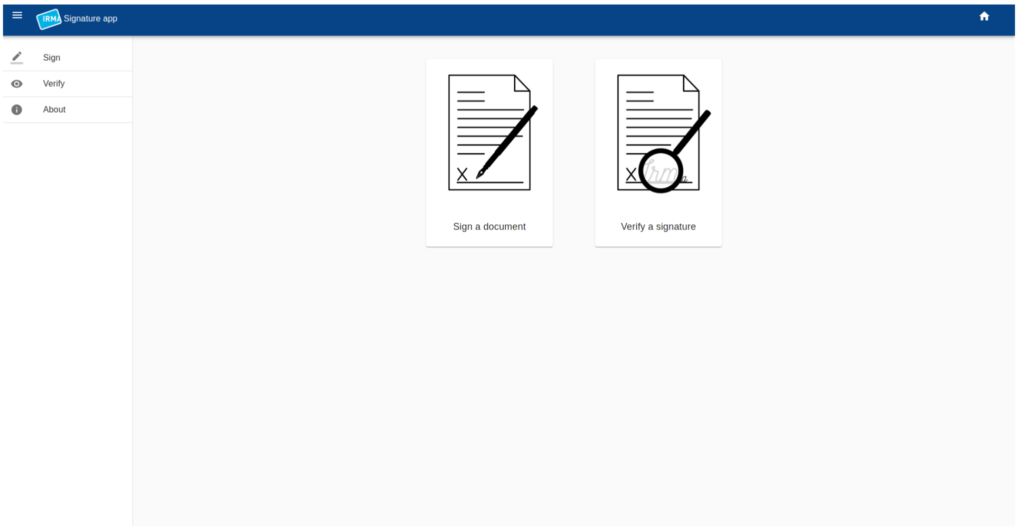
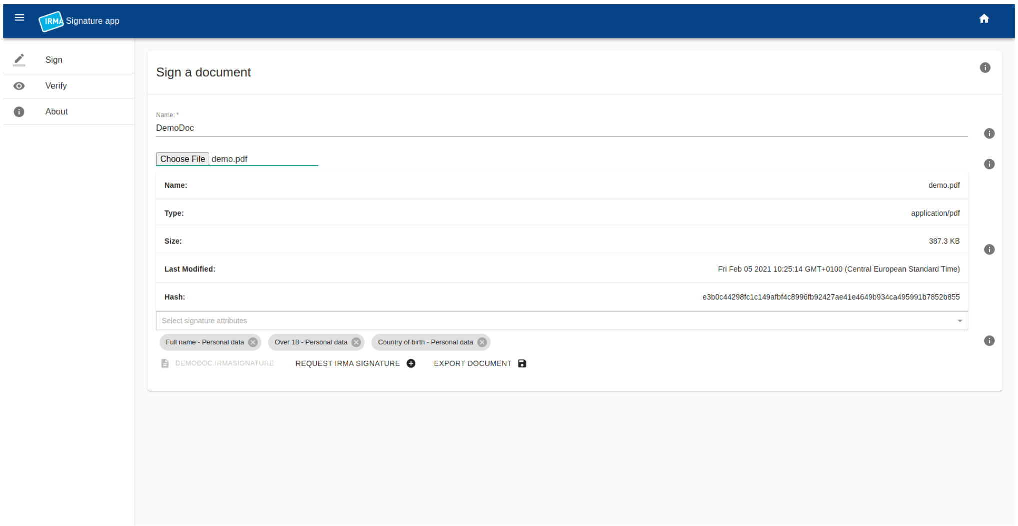


Figure 10: ISA Home-screen

The ‘ISA Home-screen’ acts as the starting point of the application. This is the first screen that is displayed to the user when the application is started. From here, the user can decide to select either ‘sign a document’ which directs the user to the sign page or select ‘verify a signature’ which directs the user to the verify page. Another option for the user is to use the side menu, allowing the user to be directed to the about page.

A.2 Sign



Sign a document

Name: *
DemoDoc

Choose File demo.pdf

Name:	demo.pdf
Type:	application/pdf
Size:	387.3 KB
Last Modified:	Fri Feb 05 2021 10:25:14 GMT+0100 (Central European Standard Time)
Hash:	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Select signature attributes

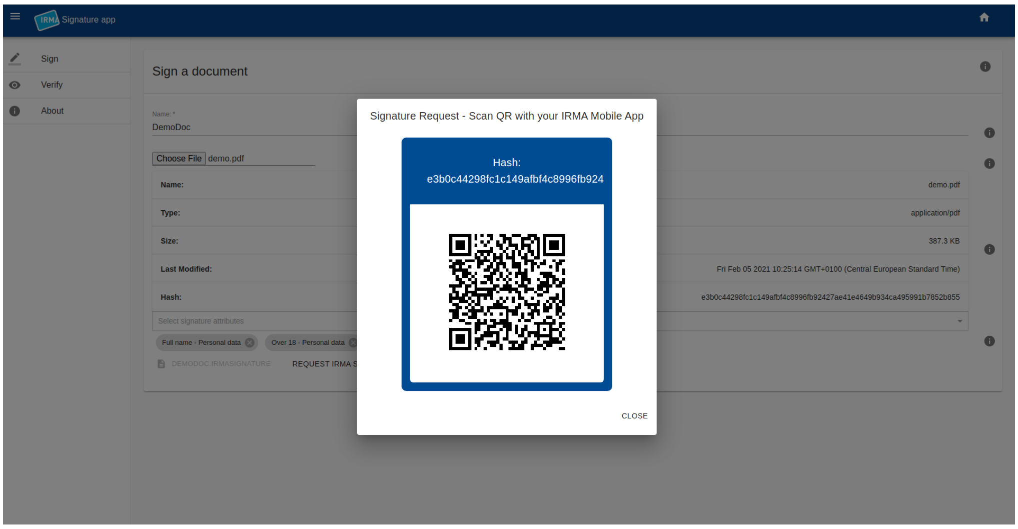
Full name - Personal data ⊗ Over 18 - Personal data ⊗ Country of birth - Personal data ⊗

DEMODOC.IRMASIGNATURE

Figure 11: ISA Sign-screen

The ‘ISA Sign-screen’ allows users to sign digital content using an IRMA attribute-based signature. The user must give the resulting (signed) file a name. Followed by choosing any file (digital content) from his local system. The application will show the metadata of the selected file. This includes the name, type, size, last modified, and hash of the file. The hash is calculated by the application based on the base64 encoding of the selected file. Followed by selecting any attributes that the user can disclose. Meaning that the user should have the attributes available in his IRMA mobile app, otherwise the signature request will fail. Once the name, file, and attributes are chosen and selected, the user can make an IRMA signature request. This will cause a popup to show, displayed in the next chapter A.3 ‘QR Scan’. Assuming the request succeeded, the user can decide to save the newly created (.irmasignature) file. This file includes both the original file as well as the IRMA signature.

A.3 QR Scan



Sign a document

Name: * DemoDoc

demo.pdf

Name:	demo.pdf
Type:	application/pdf
Size:	387.3 KB
Last Modified:	Fri Feb 05 2021 10:25:14 GMT+0100 (Central European Standard Time)
Hash:	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

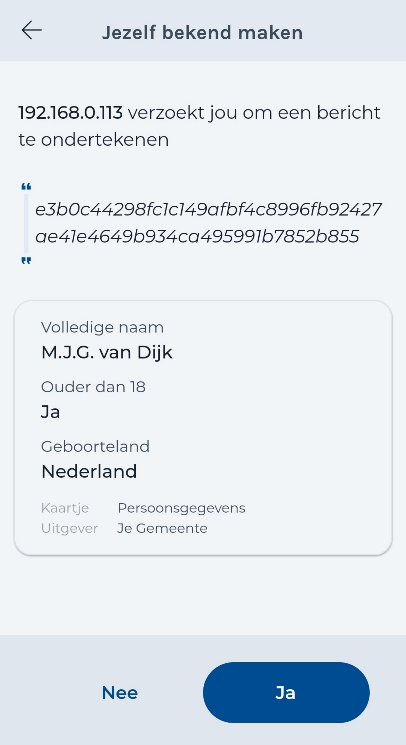
Signature Request - Scan QR with your IRMA Mobile App

Hash:
e3b0c44298fc1c149afbf4c8996fb924

Figure 12: ISA QR Scan-screen

The ‘ISA QR Scan-screen’ pops up when the user selects the ‘Request IRMA Signature’ button on the sign screen. The user can scan the QR code in order to complete the request. From here, the user is required to use the IRMA mobile app. Details about the mobile use can be seen in the next chapter A.4 ‘Mobile App’. Both on the QR scan-screen and the mobile app-screen, the hash of the original file (digital content) is shown. This allows the user to validate if the correct content (translated into a hash) is signed.

A.4 Mobile App



192.168.0.113 verzoekt jou om een bericht te ondertekenen

> *e3b0c44298fc1c149afbf4c8996fb92427 ae41e4649b934ca495991b7852b855*

Volledige naam
M.J.G. van Dijk

Ouder dan 18
Ja

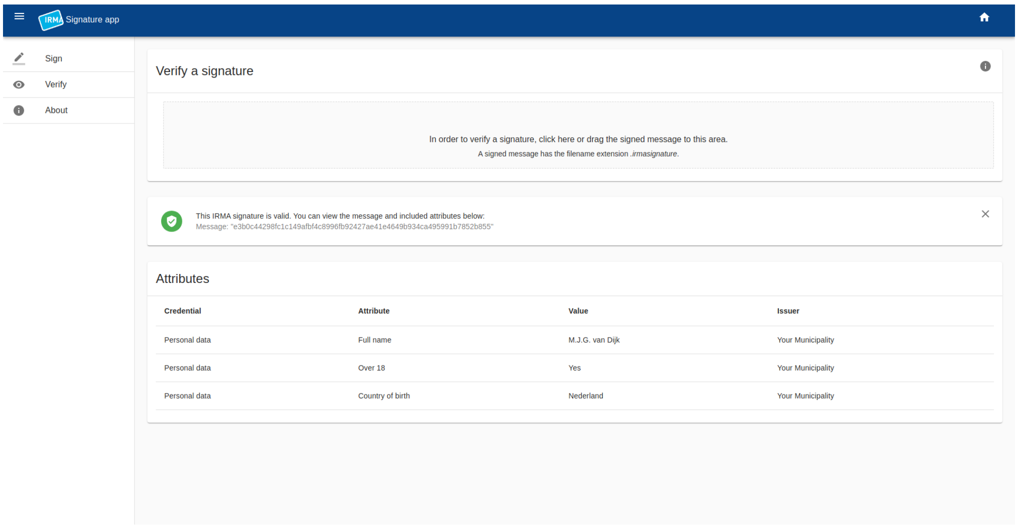
Geboorteland
Nederland

Kaartje Persoonsgegevens
Uitgever Je Gemeente

Figure 13: ISA Mobile App-screen

The ‘Mobil App-screen’ shows the user several details about what is signed. This includes the hash (as part of the message field), the attributes (within a credential), and the issuer’s name. The user can decide whether or not to sign the message (hash) with an IRMA attribute-based signature. The hash is also shown within the ISA, as can be seen on the ‘QR-screen’. The QR and the hash are shown on the computer screen. The hash shown within the mobile app can be compared by the user to the hash shown on the computer screen. Note that the user has to trust the ISA to provide the mobile app with the correct hash and show the correct hash on the ISA screens.

A.5 Verify



Verify a signature

In order to verify a signature, click here or drag the signed message to this area.
A signed message has the filename extension *.irmasignature*.

This IRMA signature is valid. You can view the message and included attributes below:
Message: "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"

Attributes

Credential	Attribute	Value	Issuer
Personal data	Full name	M.J.G. van Dijk	Your Municipality
Personal data	Over 18	Yes	Your Municipality
Personal data	Country of birth	Nederland	Your Municipality

Figure 14: ISA Verify-screen

The ‘Verify-screen’ allows the user to sign any digital content signed with an IRMA attribute-based signature using the ISA. The signed content can be selected as long as it has the correct (.irmasignature) extension. Assuming a correct file is selected, the application will show whether or not the attached IRMA signature is valid and what attributes are present within the signature. As shown in figure 14 this can include attributes like the user’s full name, whether the user is over 18, and the country of birth of the user. Additionally, the credential is displayed in which the related attribute was present, and the name of the issuer of the related attribute is displayed. Finally, the hash (the message that is signed) is displayed for the user to validate.