Pflichtenheft

„SecurePic“

Adaptive Communication GmbH

Projektbezeichnung	*SecurePic*			
Projektgeber	Menschenrechtsorganisation *“Amnesty International”*			
Erstellt am	11.04.2022			
Letzte Änderung am	24.04.2022			
Status	X	in Bearbeitung		fertiggestellt
Aktuelle Version	1.7			

Änderungsverlauf

Datum	Version	Geänderte Kapitel	Änderung	Autor	Status
11.04.2022	1.0	Alle	Erstellung	Alle	i. B.
13.04.2022	1.1	Alle	Bearbeitung	Alle	i. B.
15.04.2022	1.2	Alle	Bearbeitung	FW, JL, HE, KE, TT	i. B.
18.04.2022	1.3	Alle	Bearbeitung	HE, FW, KE	i. B.
19.04.2022	1.4	Alle	Bearbeitung	FW, HE, KE, TT	i. B.
20.04.2022	1.5	Alle	Bearbeitung	Alle	i. B.
23.04.2022	1.6	Alle	Bearbeitung	FW, JL, TT, HE	i. B.
24.04.2022	1.7	Alle	Bearbeitung	TT, FW, JL, HE, KE	i. B.

Inhalt

1 Allgemeines

1.1 Glossar

Bezeichnung	Bedeutung
SecurePic	Bezeichnung der Anwendung, welche dieses Pflichtenheft beschreibt
Kunde	Menschenrechtsorganisation „*Amnesty International*"
Projektgeber	
Projektnehmer	Projekt-Team / Verfasser des Pflichtenhefts
Träger-Bild	Bild, in welches die zu übermittelnden Informationen codiert werden/wurden
codieren	Einarbeiten der Informationen in das Träger-Bild
decodieren	Auslesen der Informationen, die in dem Träger-Bild codiert wurden
verschlüsseln	Umwandeln der zu übertragenden Information mit einem Schlüssel/Passwort, um dessen Inhalt unkenntlich zu machen[1]
entschlüsseln	Informationen „mithilfe des bei der Verschlüsselung verwendeten Schlüssels wieder lesbar machen"[2]
F	Funktionale Anforderungen
NF	Nicht-Funktionale Anforderungen
TA	Technische Anforderung
LOG	Anforderungs-Bereich Systemlogik/Allgemeine Anforderungen
GUI	Anforderungs-Bereich Grafische Benutzeroberfläche
TEC	Anforderungsbereich technische Sicht
SYS	Anforderungsbereich systemische Sicht

[1] Vgl. https://www.duden.de/rechtschreibung/verschluesseln [15.04.2022]
[2] siehe https://www.duden.de/rechtschreibung/entschluesseln [15.04.2022]

1.2 Team

Rolle(n)	Name	E-Mail
Projektleiter	Kirolis Eskondis	s200289@student.dhbw-mannheim.de
Communication Manager	Thu Giang Tran	s200307@student.dhbw-mannheim.de
Technical Manager	Frederik Wolter	s200312@student.dhbw-mannheim.de
Mitarbeiter Nr.1	Hassan El-Khalil	s200286@student.dhbw-mannheim.de
Mitarbeiter Nr.2	Kai Schwab	s200304@student.dhbw-mannheim.de
Mitarbeiter Nr.3	Jonas Lauschke	s200297@student.dhbw-mannheim.de

1.3 Einleitung

Das Internet und besonders die sozialen Medien werden zur Kommunikation, Selbstdarstellung, Verbreitung von Ansichten und auch als Informations- und Nachrichtenquelle genutzt. Private wie auch öffentliche Personen können beispielsweise auf den Plattformen sozialer Medien Inhalte in Form von Text-, Foto- oder Videobeiträgen hochladen und mit großer Reichweite schnell verbreiten.

Für vereinzelte Regierungen und Institutionen stellt diese Möglichkeit der freien öffentlichen Meinungsäußerung eine Bedrohung dar. So können beispielsweise über das Internet veröffentlichte regierungsfeindliche Informationen das Meinungsbild der Bevölkerung beeinflussen und die Stellung der Machthaber schwächen.

Außerdem geben Menschen für gewöhnlich online viele Informationen über sich und ihre Umgebung preis, was das Internet zu einer lukrativen Quelle für das Sammeln von Daten macht. Jene Daten können unter anderem für das *Profiling* von Nutzern eingesetzt werden. Anhand dieser Information können Personen (-Gruppen) gezielt manipuliert oder auch verfolgt werden.

Diese sind, neben vielen weiteren, Gründe, weshalb Regierungen und andere Organisationen ein großes Interesse an der Überwachung des Internets und besonders der sozialen Medien haben. Dabei können besagte Organisationen durch das Ausüben von Druck auf die Plattformanbieter und das Ausnutzen von Hintertüren, z. T. sogar die Verschlüsselung von Privatnachrichten umgehen und den privaten Nachrichtenverkehr von Nutzern mitlesen.

Der Kunde *Amnesty International* hat das Produkt „*SecurePic*“ ausgeschrieben – eine Software zur sicheren Informationsübertragungen für Menschen, die sich für ihre Menschenrechte einsetzen und aufgrund dessen rechtlich verfolgt oder gesellschaftlich ausgegrenzt werden.

Zweck dieses Pflichtenhefts ist es die ausgeschriebene Software „*SecurePic*“ möglichst genau und nachvollziehbar zu beschreiben, um so dem Projektgeber die Auswahl eines Angebots zu ermöglichen. Außerdem dient das Pflichtenheft dazu, das fertige Produkt bei Lieferung auf dessen Einhalten der vereinbarten Funktionen etc. zu überprüfen.

2 Konzept

2.1 Problemstellung & Ziel

Wie bereits eingangs beschrieben, ist es bekannt, dass in Teilen der Welt die sozialen Medien und digitale Kommunikation abgehört und überwacht werden. Dies führt zu Einschränkungen der Meinungsfreiheit und zu Zensur, sowie evtl. Verfolgung und Bestrafung von Personen, die sich nicht an jene halten. Die **Zielgruppe** von *SecurePic* sind Menschen (-Gruppen), die in ihrer Meinungsfreiheit eingeschränkt bzw. die Folgen ihrer öffentlichen Meinungsäußerung fürchten. So gehören zur Zielgruppe der Software unter anderem soziale Randgruppen und Minderheiten, welche sozialen Stigmata ausgesetzt sind und aufgrund dessen in der Gesellschaft ausgegrenzt oder sogar unterdrückt werden.

Der sonst übliche Weg, Informationen vor dem Zugriff Dritter zu schützen, ist die **Verschlüsselung** der Informationen. Doch verschlüsselte Nachrichten können in diesem Fall nur bis zu einem gewissen Grad helfen. Denn verschlüsselte Nachrichten werden schnell als solche erkannt und können, je nach Ausmaß der Freiheitseinschränkung, zu Verdacht, strengerer Überwachung oder gar Verfolgung führen. Sobald der Verdacht auf eine verschlüsselte Nachricht besteht, kann außerdem versucht werden, diese mit entsprechenden Algorithmen zu entschlüsseln oder Hintertüren auf den Endgeräten zu nutzen.

Für eine sichere Kommunikation müssen Nachrichten auf eine Weise übermittelt werden, die für Drittpersonen nicht auffällig erscheint. Die Nutzung von **Steganographie in Kombination mit Verschlüsselung** bietet sich in einem solchen Fall an. Steganographie beschreibt das Verstecken von Information in einem Medium. „*SecurePic*“ soll den Nutzern die Möglichkeit geben, Informationen in einem Bild zu codieren. Nachrichten in Form von Texten oder Bildern werden auf einem Träger-Bild versteckt. Einer Drittperson erscheint die Kommunikation nur wie ein Austausch von Bildern zwischen Nutzern.

Das Ziel ist, dem Anwender einen sicheren Weg zur Kommunikation zu ermöglichen, wodurch er Überwachung entgehen und sein Recht auf Meinungsfreiheit rudimentär realisieren kann.

2.2 Einsatzbereich

Die Software wird so entwickelt, dass sie als ein Werkzeug für die Gewährleistung eines sicheres Kommunikationskanals auf den *Personal Computers* der Nutzer eingesetzt werden kann. Mit Hilfe der Software soll dem Nutzer die Möglichkeit gegeben werden, Informationen in das Träger-Bild zu codieren und dieses im Anschluss über bereits bestehende digitale Kommunikationskanäle zu versenden. Als Empfänger eines codierten Träger-Bildes kann der Nutzer dann die versteckte Information mit Hilfe von SecurePic decodieren.

Eingesetzt werden soll *SecurePic* in allen Teilen der Welt, in denen befürchtet werden muss, aufgrund seiner Meinung staatlich oder gesellschaftlich benachteiligt zu werden. Gerade Menschen, die sich gegen unterdrückende Gruppierungen und Regierungen stellen, um sich für ihre Menschenrechte einzusetzen, sind besonders gefährdet und sollen mit Hilfe der Software in ihren Tätigkeiten unterstützt werden. Die effiziente, sichere und zielgerichtete Verteilung der Software übernimmt hierbei der Projektgeber.

Im Folgenden wird ein erster Entwurf der beschriebenen Software dargestellt. Die Software hat dabei die folgenden identifizierten Use-Cases:

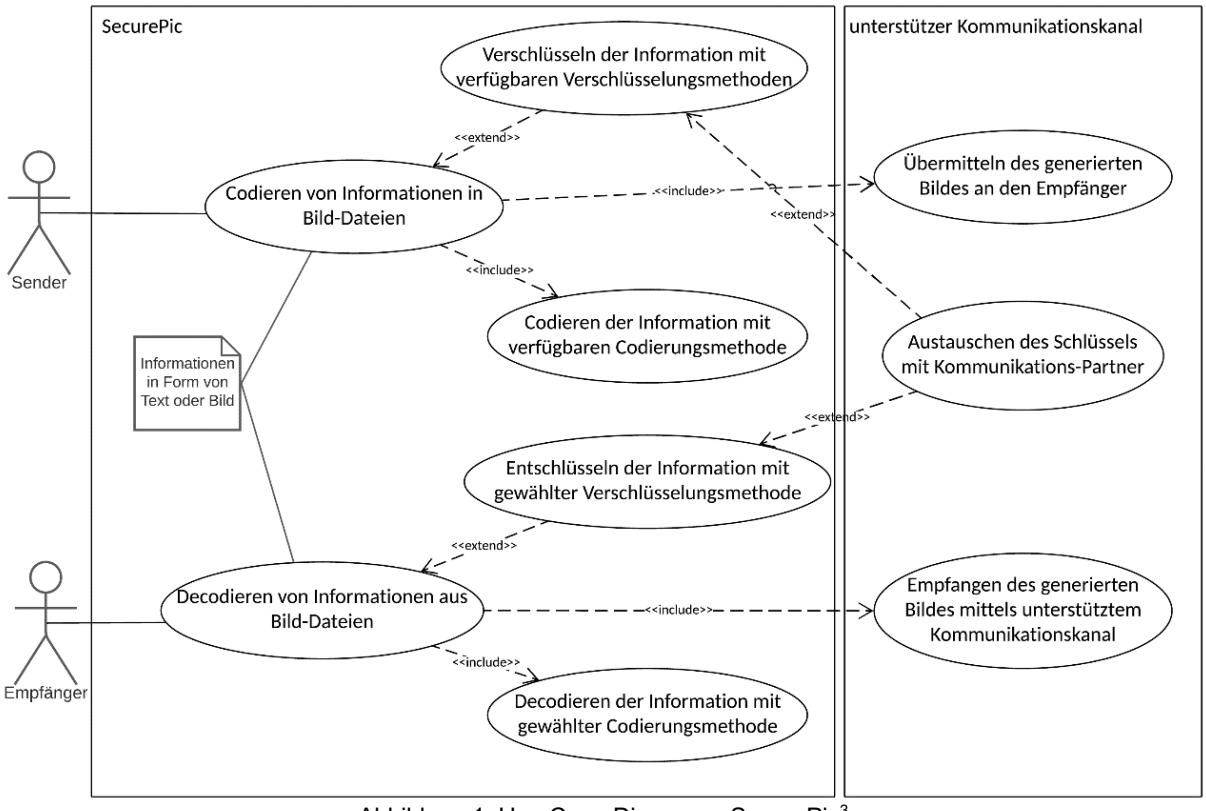

Abbildung 1: Use-Case-Diagramm SecurePic[3]

[3] Abbildung 1 eigene Darstellung

In Abbildung 1 ist zu sehen, dass es zwei Gruppen von Nutzern der Software gibt, der *Sender* sowie der *Empfänger*. Für eine Kommunikation ist ein Sender und mindestens ein Empfänger notwendig. Der Sender codiert mit Hilfe von SecurePic die zu übermittelnde Information in ein Träger-Bild seiner Wahl. Hierfür stehen dem Sender verschiedene Codierungs-Algorithmen sowie eine optionale zusätzliche Verschlüsselung der Information zur Verfügung. Der Empfänger kann mit Hilfe von SecurePic empfangene Träger-Bilder decodieren und nach evtl. Eingabe des Passworts/Schlüssels die Information zurückgewinnen.

Abbildung 2 und Abbildung 3 stellen noch einmal den Workflow des Codierens bzw. Decodierens von Informationen mit Hilfe von SecurePic in einem Flussdiagramm dar.

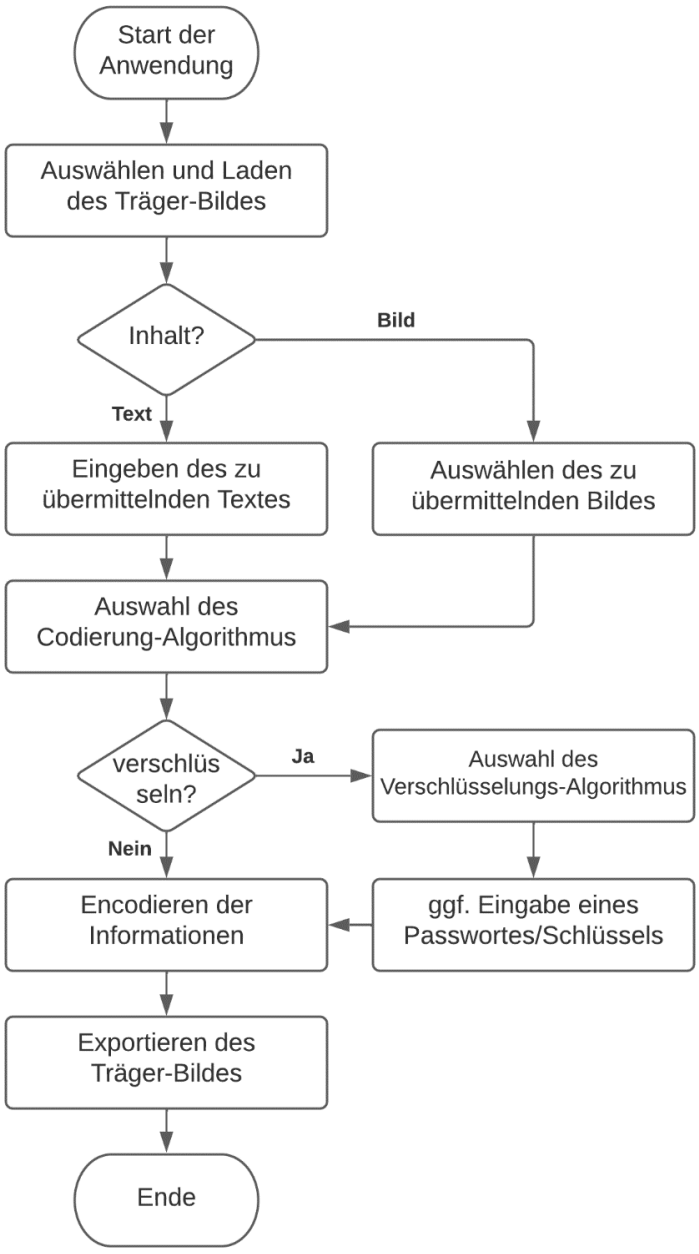

Abbildung 2: Flussdiagramm für das Codieren von Informationen (Sender-Seite)[4]

[4] Abbildung 2 eigene Darstellung

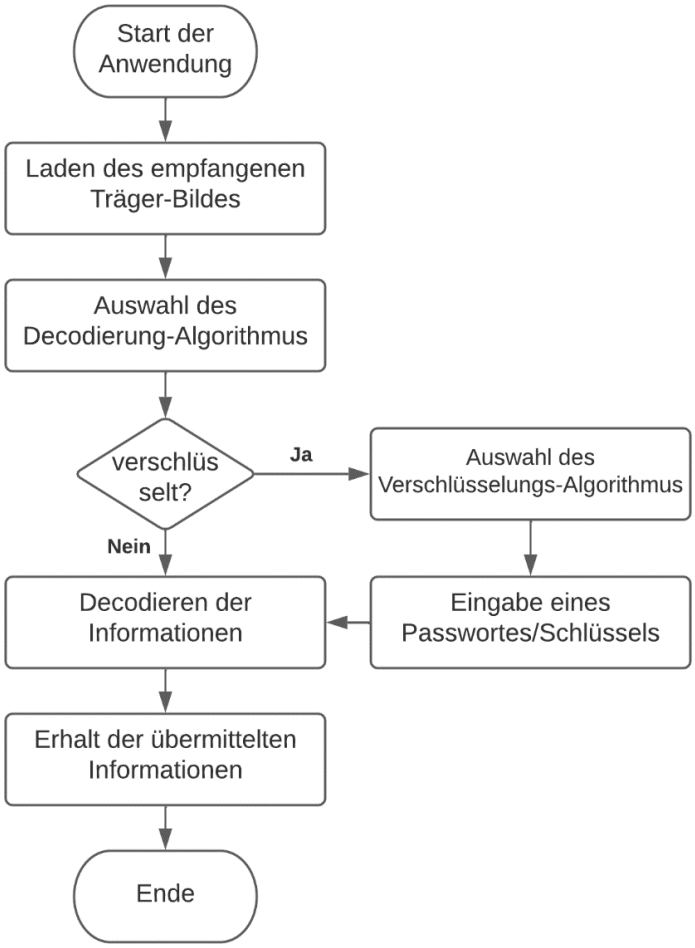

Abbildung 3: Flussdiagramm für das Decodieren von Informationen (Empfänger-Seite)[5]

Nachfolgend werden unterschiedliche Use-Cases genauer erläutert. Hierbei wird darauf geachtet die Erklärungen unabhängig voneinander zu vollziehen, um den Use-Case eigenständig verständlich zu halten. Dabei werden bewusst redundante Formulierungen verwendet.

2.2.1 Use Case: Direktnachricht (1 zu 1) ohne Verschlüsselung

Im Folgenden wird die eins-zu-eins Kommunikation zweier Anwender mittels *SecurePic* ohne Verschlüsselung erklärt.

Um eine versteckte Nachricht zu versenden, wählt der Sender ein Träger-Bild aus und übergibt es der Software, zusammen mit der zu codierenden Nachricht. Bei Bestätigung generiert die Software ein visuell identisches Bild, welches der Anwender exportieren kann. Dieses Bild kann im Anschluss über einen unterstützten Kommunikationskanal an den Empfänger versandt werden.

Der Empfänger kann das erhaltene Bild in die Software laden, um die codierte Information in der Software anzeigen zu lassen. War keine Nachricht in dem Träger-Bild, wird dem Nutzer eine inkohärente Zeichenkette bzw. eine Fehlermeldung angezeigt.

[5] Abbildung 3 eigene Darstellung

2.2.2 Use Case: Direktnachricht (1 zu 1) mit Schlüsselaustausch (symmetrisch)

Im Folgenden wird die eins-zu-eins Kommunikation zweier Anwender mittels *SecurePic* mit Schlüsselaustausch erklärt. Um eine Kompromittierung des Kommunikationskanals, zum Beispiel in Form eines „Man-in-the-Middle“ Angriffs, auszuschließen, tauschen die Anwender über einen gesicherten Kommunikationskanal, wie ein persönliches Treffen, im Vorfeld ein Passwort aus, welches Sie für die Kommunikation verwenden.

Um eine versteckte Nachricht zu versenden, wählt der Sender ein Träger-Bild aus und übergibt es der Software, zusammen mit der zu codierenden Nachricht und dem zuvor vereinbarten Passwort. Bei Bestätigung generiert die Software ein visuell identisches Bild, welches der Anwender exportieren kann. Dieses Bild kann im Anschluss über einem unterstützten Kommunikationskanal an den Empfänger versandt werden.

Der Empfänger kann das erhaltene Bild, zusammen mit dem vereinbarten Passwort, an die Software übergeben, um die codierte Information in der Software anzeigen zu lassen. War keine Nachricht in dem Träger-Bild, wird dem Nutzer eine inkohärente Zeichenkette bzw. eine Fehlermeldung angezeigt. Der Ablauf einer Kommunikation mittels symmetrischer Verschlüsselung wird in Abbildung 4 nocheinmal verdeutlicht:

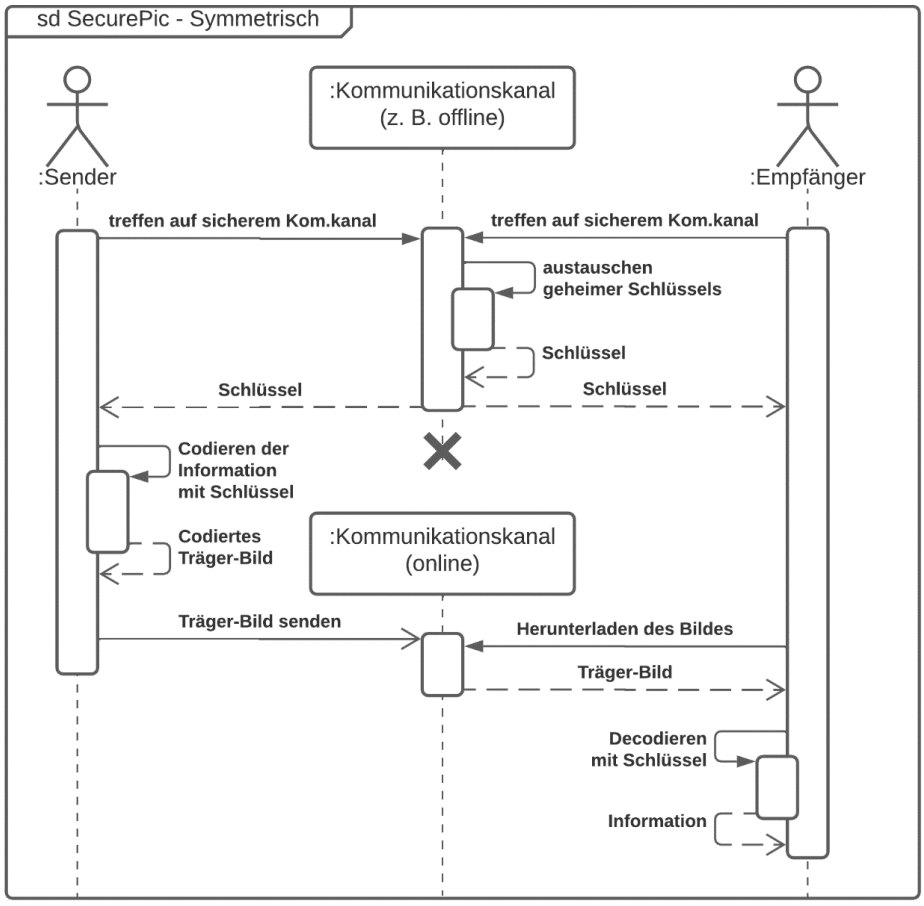

Abbildung 4: Sequenz-Diagramm Informationsaustausch mit symmetrischer Verschlüsselung[6]

[6] Abbildung 4 eigene Darstellung; entspricht nicht dem UML-Standard, dient lediglich der Veranschaulichung

2.2.3 Use Case: Direktnachricht (1 zu 1) ohne Schlüsselaustausch (asymmetrisch)

Im Folgenden wird die eins-zu-eins Kommunikation zweier Anwender mittels SecurePic ohne persönlichen Schlüsselaustusch, mit asymmetrischer Verschlüsselung, erklärt. Asymmetrische Verschlüsselungsverfahren arbeiten hierbei für gewöhnlich mit zwei Schlüsseln – dem *öffentlichen* Schlüssel und dem *privaten* Schlüssel. Mit dem öffentlichen Schlüssel verschlüsselte Daten können nur mit dem privaten Schlüssel entschlüsselt werden.

Um eine versteckte Nachricht zu versenden, generiert der Empfänger zuerst mit Hilfe von SecurePic einen öffentlichen und einen privaten Schlüssel. Der öffentliche Schlüssel wird dann unverschlüsselt in ein Träger-Bild codiert und an den Sender übermittelt. Mithilfe des öffentlichen Schlüssels codiert der Sender die zu übermittelnde Information in ein Träger-Bild und sendet es zurück an den Empfänger.

Der Empfänger kann nun das erhaltene Bild, zusammen mit dem privaten Schlüssel, an die Software übergeben, um die codierte Information in der Software anzeigen zu lassen. War keine Nachricht in dem Träger-Bild, wird dem Nutzer eine inkohärente Zeichenkette bzw. eine Fehlermeldung angezeigt. Der Ablauf einer Kommunikation mittels asymmetrischer Verschlüsselung wird in Abbildung 5 nocheinmal verdeutlicht:

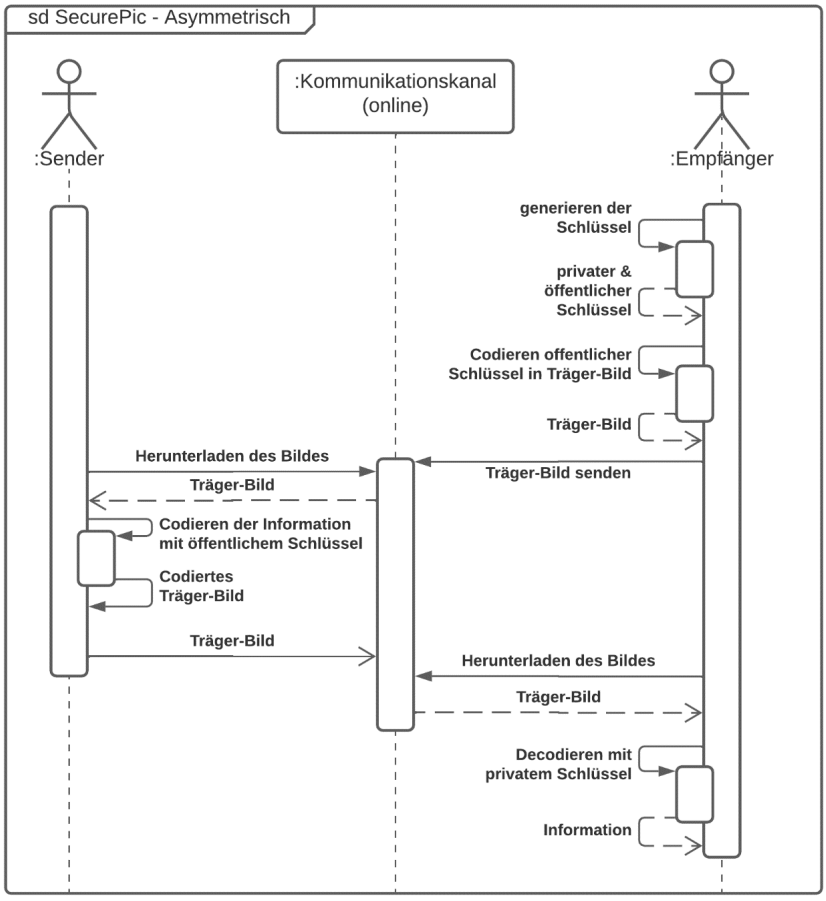

Abbildung 5: Sequenz-Diagramm Informationsaustausch mit asymmetrischer Verschlüsselung[7]

[7] Abbildung 5 eigene Darstellung; entspricht nicht dem UML-Standard, dient lediglich der Veranschaulichung

2.2.4 Use Case: Broadcast auf einem Imageboard

Im Folgenden wird die Kommunikation eines Senders und mehrerer Empfänger mittels *SecurePic* erklärt. Um eine Kompromittierung der Vertraulichkeit der Kommunikation auszuschließen, wird über einen externen gesicherten Kommunikationskanal, wie persönliche Treffen, ein Passwort vom Sender an alle Empfänger distribuiert.

Um eine versteckte Nachricht zu versenden, wählt der Sender ein Träger-Bild aus und übergibt es der Software, zusammen mit der zu versteckenden Nachricht und dem zuvor verteilten Passwort. Bei Bestätigung generiert die Software ein visuell identisches Bild, welches der Sender auf einer unterstützten öffentlichen Plattform, wie einem Imageboard, veröffentlichen kann, wo dieses den Empfängern zugänglich ist.

Die Empfänger können, die vom Sender veröffentlichten Bilder zusammen mit dem vom Sender verteilten Passwort, an die Software übergeben, um die codierte Information in der Software anzeigen zu lassen. War keine Nachricht in dem Träger-Bild, wird dem Nutzer eine inkohärente Zeichenkette bzw. eine Fehlermeldung angezeigt.

2.2.5 Use Case: Physisches Medium

Im Folgenden wird die Kommunikation mehrerer Anwender mittels *SecurePic* über ein physisches Medium, wie einen USB-Stick oder eine CD, erklärt. Um eine Kompromittierung des Kommunikationskanals nach Schlüssel-Austausch auszuschließen, tauschen die Anwender über einen gesicherten Kommunikationskanal, wie ein persönliches Treffen, im Vorfeld ein Passwort aus, welches Sie für die Kommunikation verwenden.

Um eine versteckte Nachricht zu versenden, wählt der Sender ein Träger-Bild aus und übergibt es der Software, zusammen mit der zu versteckenden Nachricht und dem zuvor vereinbarten Passwort. Bei Bestätigung generiert die Software ein visuell identisches Bild, welches der Benutzer exportieren kann. Dieses Bild kann dann vom Nutzer auf das physische Medium transferiert zu werden, welches er dann an seine Kommunikationspartner verteilt.

Die Empfänger können die vom Sender erhaltenen Bilder, zusammen mit dem vom Sender verteilten Passwort, an die Software übergeben, um die codierten Informationen in der Software anzeigen zu lassen. War keine Nachricht in dem Träger-Bild, wird dem Nutzer eine inkohärente Zeichenkette bzw. eine Fehlermeldung angezeigt.

2.3 Mockup

Ein erster Entwurf für die Benutzeroberfläche der Software sieht wie folgt aus:

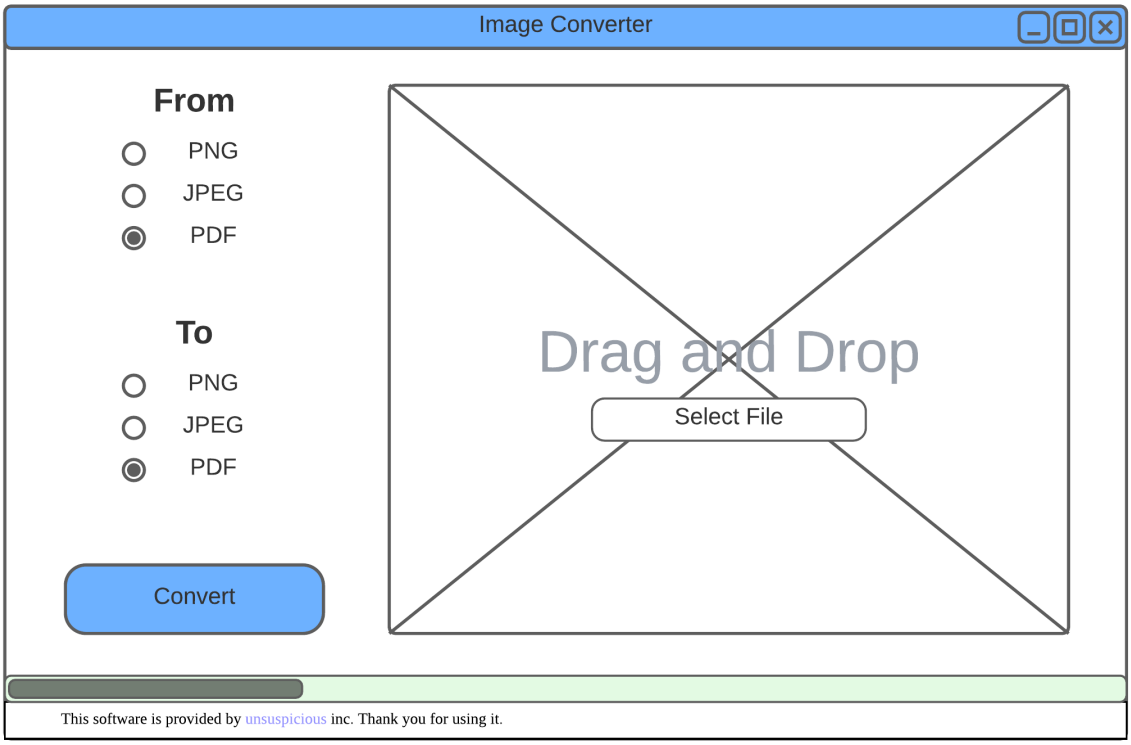

Abbildung 6: Mockup „Image-Converter“ Tarn-Ansicht[8]

In Abbildung 6 ist zu erkennen, dass nach dem Starten der Software die Applikation zunächst als „Image-Converter“ geöffnet wird. Im Falle einer Untersuchung des Anwendergeräts, soll diese Tarnung helfen die Software nicht verdächtig wirken zulassen. Mit einem Klick auf das hervorgehobene Wort in der Fußzeile, wird die eigentliche Applikation *SecurePic* geöffnet:

[8] Abbildung 6 eigene Darstellung

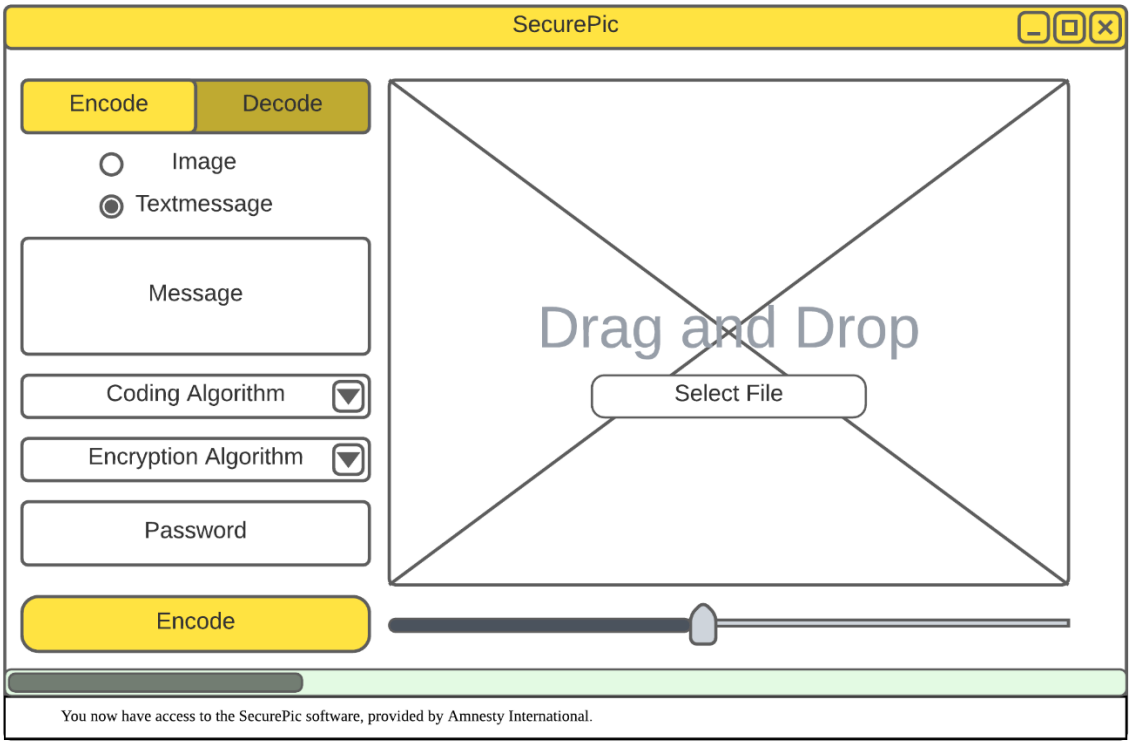

Abbildung 7: Mockup Codierungs-Ansicht von SecurePic[9]

Wie in Abbildung 7 dargestellt wird, gibt es zwei Ansichten – die Codierungs- und Decodierungs-Ansicht – die durch den Nutzer ausgewählt werden können.

In der **Codierungs-Ansicht** kann zunächst ausgewählt werden, in welcher Form die zu sendende Information vorliegt. Darunter folgt ein Eingabefenster für die Information. Anschließend kann in einem Dropdown der Codierungs-Algorithmus für das Bild und darunter der Verschlüsselungs-Algorithmus ausgewählt werden. Beim Verschieben des Reglers kann eingestellt werden, wie stark die Qualität des Bildes beim Codieren in das Träger-Bild verändert wird. Zum Schluss wird durch das Drücken des *Encode-Buttons* ein visuell identisches Bild von der Software generiert, das anschließend vom Anwender exportiert werden kann.

[9] Abbildung 7 eigene Darstellung

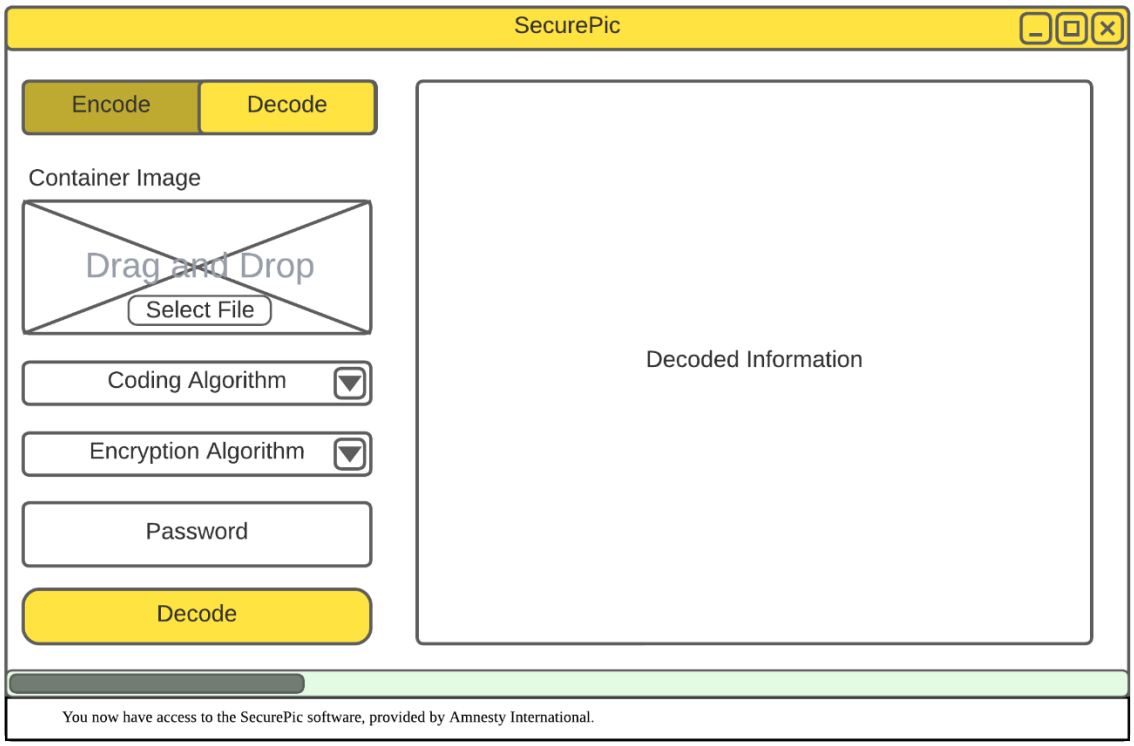

Abbildung 8: Mockup Decodierungs-Ansicht von SecurePic[10]

Wie in Abbildung 8 visualisiert, lässt sich in der **Decodierungs-Ansicht** das Trägerbild mittels Drag & Drop oder einem „File-Select"-Dialog in die Software laden. Anschließend kann in einem Dropdown der Verschlüsselungs-Algorithmus und der Codierungs-Algorithmus ausgewählt werden mit denen das Träger-Bild codiert wurde. Je nach Auswahl ist es möglich das Passwort bzw. den Schlüssel einzugeben. Zum Schluss wird durch das Drücken des *‚Decode'-Buttons* die codierte Information des Träger-Bildes in dem rechten Ausgabefenster angezeigt.

[10] Abbildung 8 eigene Darstellung

3 Anforderungen

3.1 Funktionale Anforderungen

Definition der Prioritäten: 1 = muss; 2 = soll; 3 = kann

ID	Anforderungen	Abhängig.	Prio
FA-LOG-00	Die Software ermöglicht einen Kommunikationskanal für die sichere Meinungsäußerung und Informationsaustausch.		1
FA-LOG-10	Die Software ermöglicht eine versteckte Kommunikation über ein kompromittiertes Medium.		1
FA-LOG-10.1	Die Software selbst verfügt nicht über die Funktion Nachrichten/Informationen zu übermitteln bzw. zu versenden.	FA-LOG-10.2	1
FA-LOG-10.2	Die Software ermöglicht das Verwenden bestehender unterstützter Kommunikationskanäle für den Informationsaustausch.		1
FA-LOG-20	Die Software ermöglicht den Informationsaustausch mittels versteckter Informationen in **digitalen Bild-Dateien**.		1
FA-LOG-20.1	Die Software ermöglicht das **Übermitteln rudimentärer Informationen**, welche nachfolgend genauer spezifiziert werden.		1
FA-LOG-20.1.1	Die Software ermöglicht das Übermitteln von versteckten Informationen in Form von **Text**.		1
FA-LOG-20.1.2	Die Software ermöglicht das Übermitteln von versteckten Informationen in Form von **Bild-Dateien**.		1
FA-LOG-20.1.3	Die Software ermöglicht das Übermitteln von versteckten Informationen in Form von **Audio-Dateien**.		3
FA-LOG-20.2	Die Software ermöglicht das **Auslesen versteckter rudimentärer Informationen**, welche durch eine andere Instanz der Software versteckt wurden.	FA-LOG-20.1	1
FA-LOG-20.2.1	Mehrere Instanzen der Software können die gleiche Information aus derselben Bild-Datei empfangen (Broadcast-Funktion).		2
FA-LOG-20.3	Die Software unterstützt das Übertragen rudimentärer Nachrichten zwischen verschiedenen Instanzen der Software.	FA-LOG-20.1, FA-LOG-20.2	1
FA-LOG-30	Die Software ermöglicht den Informationsaustausch mittels versteckter Informationen in digitalen **Bildern**, welche **ausgedruckt** wurden.	FA-LOG-20	3
FA-LOG-40	Versteckte Informationen können optional vor unbefugtem Zugriff durch **Verschlüsselung** gesichert werden.		1
FA-LOG-50	Die Software ermöglicht, in Abhängigkeit des vom Nutzer gewählten Codierungsalgorithmus und anderer Einstellungen, mindestens* nachfolgende Codierungseffizienz. *Die Werte beziehen sich auf unkomprimierte Bilder mit einer min. Größe von 512x512 Pixel. Sie beschreiben den Anteil der "codierbaren" Daten-Größe im Vergleich zu der		1

	Gesamtgröße des Träger-Bildes nach dem Codierungsvorgang.		
FA-LOG-50.1	Bei der Codierung mit dem LSB-Algorithmus ist unter optimalen Bedingungen von einer minimalen Codierungs-Kapazität von 8% auszugehen.	FA-TEC-10.1.1	1
FA-TEC-00	Die Software ermöglicht das Codieren und Decodieren von Informationen in digitalen Bild-Dateien (Träger-Bildern) mittels Steganographie und Verschlüsselung.		1
FA-TEC-00.1	Die Software speichert grundsätzlich keine Nutzerdaten, Nutzereingaben und zu verschlüsselnden Informationen, ohne dass der Nutzer dies verlangt.		1
FA-TEC-10	Die Software ermöglicht das **Codieren** von Informationen in ein digitales Trägerbild.	FA-LOG-20	1
FA-TEC-10.1	Die Software bietet verschiedene **Codierungs-Algorithmen** für die Codierung der Informationen in das Trägerbild.		1
FA-TEC-10.1.1	Least-Significant-Bit-Codierung (LSB)		1
FA-TEC-10.1.2	Plus-Minus-One-Codierung (PM1)	FA-TEC-10.1.1	2
FA-TEC-10.1.3	Diskrete Cos-Transformation-Manipulation		2
FA-TEC-10.1.4	weitere Algorithmen - Vorstudie ausstehend ##		3
FA-TEC-10.2	Träger-Bilder in die Informationen codiert wurden sind optisch mit dem bloßen Auge nicht von Bildern ohne Codierte Information zu unterscheiden*. *Je nach Einstellungen des Nutzers sind Artefakte der Codierung erkennbar.		1
FA-TEC-10.2.1	Codierte Trägerbilder können nicht mit dem bloßen menschlichen Auge von dem Ausgangsbild in originaler Vergrößerung unterschieden werden*. *Je nach Einstellungen des Nutzers sind Artefakte der Codierung erkennbar.		2
FA-TEC-10.3	Jede Information-Menge soll in jedes Träger-Bild codiert werden können, unabhänig von der eigentlichen "Kapazität" des Träger-Bildes. Hierbei kann es zu notwendigen Anpassungen der Bildgröße kommen.		3
FA-TEC-20	Die Software ist in der Lage optional die zu codierenden Informationen zu **verschlüsseln**.	FA-LOG-40	1
FA-TEC-20.1	Die Software bietet verschiedene Verschlüsselungs-Algorithmen für die Verschlüsselung der Informationen.		1
FA-TEC-20.1.1	Die Software bietet eine **symmetrische** Verschlüsselung mittels eines Vereinbarten Passwortes an.		1
FA-TEC-20.1.2	Die Software bietet eine **asymmetrische** Verschlüsselung mittels eines ähnlichen Verfahrens zu RSA an.		2
FA-TEC-30	Die Software bietet die Möglichkeit das Trägerbild nach der Codierung zu **exportieren**.	FA-LOG-10	1
FA-TEC-30.1	Die Software bietet die Möglichkeit das Trägerbild in die Zwischenablage des Betriebssystems zu kopieren.		2
FA-TEC-30.2	Die Software bietet die Möglichkeit das Trägerbild in eine Datei auf das lokale Dateisystem zu exportieren.		1

FA-TEC-30.2.1	Die Software bietet die Möglichkeit das Trägerbild in Form einer **PNG**-Datei zu exportieren		1
FA-TEC-30.2.2	Die Software bietet die Möglichkeit das Trägerbild in Form einer **JPG**-Datei zu exportieren.		1
FA-TEC-40	Die Software ermöglicht das **Decodieren** von Informationen in einem digitalen Trägerbild, welches durch eine Instanz der Software codiert wurde.		1
FA-TEC-40.1	Die Software kann Informationen aus einem Trägerbild decodieren, wenn sie mit den Einstellungen der codierenden Instanz übereinstimmt.		1
FA-TEC-40.2	Die codierten Informationen können verlustfrei decodiert werden.		3
FA-TEC-40.3	Die Software ist in der Lage die codierten Informationen zu entschlüsseln, falls die hierfür nötigen Daten, wie Schlüssel oder Passwort, vorliegen.		1
FA-TEC-40.4	Die Software ist in der Lage Träger-Bilder, welche nach der Codierung geringfügig verändert wurden, wie bspw. zuschneiden von Bildern, zu decodieren.		3
FA-TEC-40.5	Die Software ist in der Lage zwischen der Art der codierten Information, wie Text oder Bild, zu unterscheiden.		1
FA-TEC-50	Die Software bietet die Möglichkeit decodierte Informationen, nach dem Decodierungsvorgang, zu **exportieren**.		1
FA-TEC-50.1	Die Software bietet die Möglichkeit die decodierten Informationen in die Zwischenablage des Betriebssystems zu kopieren.		2
FA-TEC-50.2	Die Software bietet die Möglichkeit decodierte Informationen in Form einer Datei zu exportieren.	FA-TEC-50	1
FA-TEC-50.2.1	Die Software kann textuelle decodierte Informationen in Form von TXT-Dateien exportieren.	FA-TEC-40.5	2
FA-TEC-50.2.2	Die Software kann decodierte Bilder in Form von JPG-Dateien exportieren.	FA-TEC-50.2, FA-TEC-40.5	1
FA-TEC-60	Die Software baut keine Internetverbindung während der Verwendung auf.		1
FA-GUI-00	**Die Software besitzt eine graphische Oberfläche.**		1
FA-GUI-00.1	Die Software bietet die Möglichkeit über Tabs zwischen Codieren und Decodieren zu wechseln.		1
FA-GUI-00.2	Die GUI der Software verfügt über verständliche Beschriftung und Layout der Bedienelemente.		1
FA-GUI-00.2.1	Die GUI steht in verschiedenen Sprachen zu Verfügung.		2
FA-GUI-00.3	Das Anwendungsfenster ist skalierbar.		2
FA-GUI-20	Die Software verfügt über Bedien-Möglichkeiten der **Codierungs-Funktion**.		1
FA-GUI-20.1	Im Codieren-Tab existiert ein Feld, indem man per Drag und Drop das Träger-Bild laden kann.		1

FA-GUI-20.2	Im Codieren-Tab gibt es die Möglichkeit zwischen der Art der zu codierenden Information auszuwählen.	FA-LOG-20.1	1
FA-GUI-20.3	Im Codieren-Tab existiert ein Textfeld für die Eingabe der zu codierenden Nachricht.	FA-GUI-20.2	1
FA-GUI-20.4	Im Codieren-Tab existiert ein "File-Upload"-Feld, in welchem man das zu codierende Bild einlesen kann.	FA-GUI-20.2	1
FA-GUI-20.5	Im Codieren-Tab existiert ein Drop-Down für die Wahl des Codierungs-Algorithmus.		1
FA-GUI-20.6	Im Codieren-Tab existiert ein Drop-Down für die Wahl des Verschlüsselungs-Algorithmus.		1
FA-GUI-20.7	Im Codieren-Tab existiert ein Textfeld zur Passwort-/Schlüssel-Eingabe.	FA-GUI-20.6	1
FA-GUI-20.8	Im Codieren-Tab existiert ein Knopf zum Starten des Codierungs-Prozesses.		1
FA-GUI-20.10	Im Codieren-Tab existieren weitere Einstellmöglichkeiten, welche bspw. für die Einstellung der Kompressions-Sicherheit und "Stärke" der Bild-Manipulation genutzt werden.		3
FA-GUI-20.11	Im Codieren-Tab wird das geladene Trägerbild, in das die Informationen codiert werden sollen, angezeigt.		1
FA-GUI-20.12	Im Codieren-Tab wird die Informationskapazität des Trägerbilds angezeigt.		3
FA-GUI-30	Die Software verfügt über Bedien-Möglichkeiten der **Decodierungs-Funktion**.		1
FA-GUI-30.1	Im Decodieren-Tab existiert ein Feld, indem man per Drag und Drop das Träger-Bild laden kann.		1
FA-GUI-30.2	Im Decodieren-Tab existiert ein "File-Upload"-Feld, in welchem man das Träger-Bild laden kann.		1
FA-GUI-30.3	Im Decodieren-Tab wird das geladene Trägerbild, aus dem die Informationen decodiert werden sollen, angezeigt.		3
FA-GUI-30.4	Im Decodieren-Tab wird die decodierte Information aus dem Trägerbild in einem Ausgabefeld angezeigt.		1
FA-GUI-30.5	Im Decodieren-Tab existiert ein Drop-Down für die Wahl des Verschlüsselungs-Algorithmus des Passworts.		1
FA-GUI-30.6	Im Decodieren-Tab existiert ein Drop-Down für die Wahl des Codierungs-Algorithmus.		1
FA-GUI-30.7	Im Decodieren-Tab gibt es die Möglichkeit zwischen der Art der Passworteingabe zu wechseln		2
FA-GUI-30.7.1	Im Decodieren-Tab gibt es ein Textfeld zur Passwort-Eingabe in Form eines Textes.		1
FA-GUI-30.7.2	Im Decodieren-Tab gibt es ein "File-Upload"-Feld zur Passwort-Eingabe in Form eines Bildes.		2
FA-GUI-30.8	Im Decodieren-Tab existiert ein Knopf zum Starten des Decodier-Vorgangs.		1
FA-GUI-50	Beim Start der Software wird zunächst ein **"Tarn"-Fenster** geöffnet, um die eigentliche Verwendung der Software zu verschleiern.		2

FA-GUI-50.1	Bei dem "Tarn"-Fenster handelt es sich um einen Image-Converter.		2
FA-GUI-50.2	Über einen versteckten Button gelangt der Nutzer zur eigentlichen Benutzeroberfläche der Software.		2
FA-GUI-50.3	Im "Tarn-Fenster" werden GUI-Elemente für die Bedienung des Konverters angezeigt.		2
FA-GUI-50.3.1	Im "Tarn-Fenster" gibt es eine Auswahl des Original Bild-Formates (PNG, JPG, PDF, usw.).		2
FA-GUI-50.3.2	Im "Tarn-Fenster" gibt es eine Auswahl des Ziel-Bild.-Formates (PNG, JPG, PDF, usw.).		2
FA-GUI-50.3.3	Im "Tarn-Fenster" gibt es einen Button, um Konvertieren zu starten.	FA-GUI-50.4	2
FA-GUI-50.4	Im "Tarn"-Fenster können Bilder in das angegebenen Formate konvertiert werden.		3
FA-GUI-50.5	Im "Tarn-Fenster" existiert eine Fläche, auf der per Drag und Drop das zu konvertierende Bild hinzufügt werden kann.		2
FA-GUI-50.6	Es gibt im "Tarn-Fenster" eine Fläche, um das geladene Bild anzuzeigen.		3

3.2 Nicht funktionale Anforderungen

ID	Anforderungen	Abhängig.	Prio
NF-TEC-00	Die Software ist **funktional sicher und robust**.		1
NF-TEC-10	Die Decodierung von codierten Informationen in Trägerbildern erfolgt **funktional sicher**.		1
NF-TEC-20	Im Falle eines Fehlers im Programm und der Verarbeitung von Trägerbildern wird dem Nutzer eine **Fehlermeldung** angezeigt.		1
NF-TEC-30	Die Software ist grundsätzlich **erweiterbar**.		2
NF-TEC-30.1	Die Software kann um die Verarbeitung von weiteren Bildformaten erweitert werden.		2
NF-TEC-30.2	Die Software kann um neue Verschlüsselungs- und Codierungs-Algorithmen erweitert werden.		2
NF-TEC-40	Die Software benötigt **keine Installation**. Sie kann nach der Distribution sofort ausgeführt werden.		2
NF-TEC-50	Die Codierung von Informationen erfolgt in unter 1 Minute*. *Dies gilt für eine Träger-Bild-Größe von unter einem Megabyte.		2
NF-LOG-10	Die Software erfasst keine personenbezogenen Daten.		1
NF-GUI-10	Die Nutzeroberfläche ist schlicht und übersichtlich gestaltet.		2

4 Rahmenbedingungen

Im Folgenden werden verschiedene Rahmenbedingungen und Aspekte im Zusammenhang mit der Entwicklung und dem Einsatz der Software beschrieben.

4.1 Entwicklungsumgebung

Die Software wird in der Programmiersprache Java entwickelt. Für die Entwicklung der grafischen Benutzeroberfläche wird Swing, ein GUI Toolkit von Java, verwendet. Mit Hilfe des Versionsverwaltungs-Programm Git in Verbindung mit GitHub wird der Quell-Code der Software verwaltet.

4.2 Technische Anforderungen für Betrieb der Software „SecurePic“

Die Software benötigt einige technische Anforderungen, um korrekt verwendbar zu sein. Die genannten Anforderungen befinden sich nicht im Lieferumfang von „SecurePic“ und sind vom Projektgeber sicherzustellen. Diese sind im Folgenden in tabellarischer Form aufgelistet:

ID	Anforderungen	Abhängig.	Prio
TA-00	Zur Nutzung der Software wird ein funktionsfähiger Computer benötigt.		1
TA-10	Um die Software zu installieren ist genügend Speicher erforderlich.		1
TA-20	Das ausführende Gerät benötigt einen Bildschirm, um die Graphische Nutzeroberfläche anzuzeigen und zu nutzen.		1
TA-30	Um die Software ausführen zu können muss auf dem Endgerät eine aktuelle Version des „Java Runtime Environment“ installiert sein.		1
TA-40	Zur Verwendung der Software wird keine Internetverbindung benötigt.		1

4.3 Qualität

Der Auftragnehmer verpflichtet sich, ein qualitativ hochwertiges Produkt auszuliefern. Die Qualität lässt sich an folgenden Kriterien festhalten:

- Die fertige Anwendung wird vollumfänglich getestet, es wird sichergestellt, dass alle vereinbarten Anforderungen erfüllt werden. Alle kritischen Compiler-Warnungen werden beseitigt.
- Die Entwickler orientieren sich an Clean-Code-Konventionen, um eine schnelle Entwicklung und eine leichte Wartbarkeit sicherzustellen.

4.4 Testszenarien

Um die Qualität zu sichern, werden verschiedene Tests durchgeführt:

- Zur Gewährleistung eines reibungslosen Ablaufes, wird die Software auf den Systemen des Auftragnehmers getestet – die Installation und Tests auf Auftraggeber-Seite übernimmt der Auftraggeber.
- Es ist davon auszugehen, dass Nutzer versuchen unvorhergesehene Eingaben zu tätigen. Auch ein Klicken auf nicht-Bedienflächen ist denkbar. Unabhängig von der Eingabe darf die Software nicht abstürzen oder unvorhergesehene Ausgaben tätigen.
- Die Übertragung von Informationen mit verschiedenen Träger-Bildern wird gemäß den Anforderungen getestet, um sicherzustellen, dass die Software die geforderten Übertragungs-Funktionen erfüllt.
- Für die angegebene Mindest-Informationsdichte in Träger-Bildern werden Tests durchgeführt.

5 Liefer- und Abnahmebedingungen

Die vollständig funktionale Software (siehe Prioritäten 1 & 2 der Anforderungen im Abschnitt *3.1 Funktionale Anforderungen*) wird dem Auftraggeber in Kalenderwoche **25/26 des Jahres 2022** ausgeliefert. Die Auslieferung erfolgt auf digitale Weise, anschließend kann die Software durch den Auftraggeber heruntergeladen werden. Nach der Produktübergabe und Abnahme durch den Auftraggeber endet die, durch dieses Dokument entstandene geschäftliche Beziehung.

Funktionserweiterungen und jegliche Art der Wartung und Produktinstandhaltung sind nicht Bestandteil der Geschäftsbeziehung und werden nicht vom Auftragnehmer durchgeführt. Nach Abnahme des Produktes besteht keine weitere Gewährleistung durch den Auftragnehmer.

Im Rahmen der Auslieferung übersendet der Auftragnehmer dem Auftraggeber eine Dokumentation des Produktes. Aufgrund der intuitiven Bedienbarkeit gemäß den beschriebenen Anforderungen, ist keinerlei Schulung beziehungsweise Training mit der Software seitens des Auftragnehmers vorgesehen.

Die Kosten für das Gesamtprojekt, inklusive Umsetzung aller mit dem Kunden vereinbarten Anforderungen der Prioritäten 1 und 2 (siehe *3.2 Nicht funktionale Anforderungen*), belaufen sich voraussichtlich auf insgesamt **490 Personenstunden** und damit **31.850€.**