AJAX-Anwendung zur Demonstration der visuellen Kryptographie

Studienarbeit

Arbeitsbereich Theoretische Informatik/Formale Sprachen
Wilhelm-Schickard-Institut für Informatik
Fakultät für Informations- und Kognitionswisenschaften
Universität Tübingen

von

Alexander Christ

Betreuer:
Dr. Bernd Borchert
Dr. Klaus Reinhardt

Tag der Anmeldung: 27. Juli 2007
Tag der Abgabe: 29. März 2008

Inhaltsverzeichnis

1 Einleitung

1.1 Zielsetzung

Ziel dieser Studienarbeit ist die Anfertigung einer Internet-Anwendung. Sie soll dem Benutzer Funktionen eines Malprogramms zur Verfügung stellen, wie das Zeichnen von Linien oder Buchstaben und das Einfügen von Bildern. Das vom Benutzer kreierte Bild soll nach einem visuellen Verschlüsselungsverfahren kodiert und das Ergebnis der Kodierung zur Speicherung angeboten werden.

1.2 Gliederung

Die schriftliche Ausarbeitung gliedert sich in drei Bereiche. Im ersten Teil wird das kryptographische Verfahren erläutert. Im zweiten Teil werden Funktionalität und Anwendungsmöglichkeiten der Applikation beschrieben. Der dritte Teil befasst sich mit der für die Implementierung verwendeten Technologie.

1.3 Standort

Die Anwendung ist zu finden unter

http://www-fs.informatik.uni-tuebingen.de/studdipl/christ/

Die englische Version gibt es auf

http://www-fs.informatik.uni-tuebingen.de/studdipl/christ/indexEN.php

2 Verschlüsselungsverfahren

2.1 Kodierung

Das hier verwendete Verfahren wurde 1994 von Adi Shamir und Moni Naor beschrieben [NS94]. Die zu verschlüsselnde Nachricht liegt in Form eines Schwarz-Weiß-Bildes vor. Für jedes Pixel der Nachricht werden je vier Pixel in zwei Teilbilder erzeugt, gemäß dem Schema in Abbildung 1. Es entstehen zwei Bilder, deren vertikale und horizontale Auflösung doppet so hoch ist wie die der Nachricht und deren Inhalt aus zufällig gesetzten schwarzen und weißen Pixeln besteht.

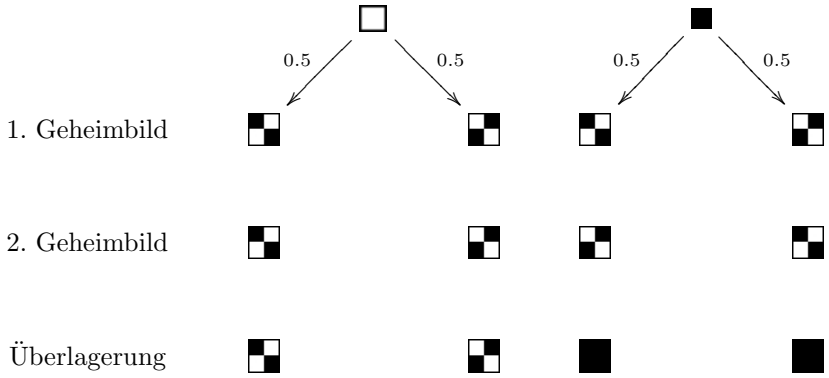

Abbildung 1: Kodierschema.

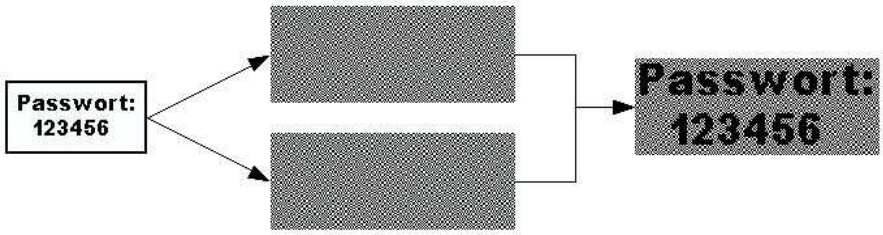

Abbildung 2: Beispiel. Links befindet sich die Nachricht, in der Mitte steht das Ergebnis der Kodierung und rechts sieht man das Resultat der Entschlüsselung.

2.2 Entschlüsselung

Die Entzifferung erfolgt durch Überlagerung der Teilbilder. Praktisch geschieht dies, indem man das eine Teilbild auf eine transparente Folie und das andere auf ein Blatt Papier druckt. Ein schwarzes Pixel der Nachricht wird in der Überlagerung durch vier schwarze Pixel repräsentiert. Ein weißes Pixel hingegen durch zwei weiße und zwei schwarze. Der Kontrast der rekonstruierten Nachricht verringert sich also um 50% im Vergleich zum Original. Dennoch sollte der Inhalt erkennbar sein. Da die Entschlüsselung allein durch visuelle Wahrnehmung geschieht spricht man von "visueller Kryptographie". Insbesondere werden also keine Computer oder kryptographische Kenntnisse vorausgesetzt.

2.3 Sicherheit

Das Verfahren ist nicht knackbar. Für jedes Pixel der Nachricht wird zufällig eine von zwei Kodierungen gewählt wird, so dass der sich Inhalt der Teilbilder nicht von einem zufälligen Rauschen unterscheidet. Ein Teilbild allein lässt keinen Rückschluss auf die ursprüngliche Nachricht zu. Auch dann nicht, wenn Vermutungen über den Inhalt der Nachricht oder große Rechenkapazitäten vorhanden sind. Damit bietet das Verfahren die kryptographische Sicherheit eines One-Time-Pads.

2.4 Variationen

Der beschriebene Algorithmus fällt in die Kategorie der (2,2)-Secret-Sharing Systeme, da eine Nachricht in zwei Teile zerlegt wird und beide für die Entzifferung notwendig sind. Shamir und Monar verallgemeinerten diesen Ansatz zu einem (n,m)-Secret-Sharing System [NS94], bei dem ein Bild in n Teile zerlegt wird und mindestens m Teile für die Entschlüsselung verfügbar sein müssen. m-1 Folien erlauben keinen Rückschluss auf die ursprüngliche Nachricht. Ein anderer Ansatz besteht darin, visueller Kryptographie mit Steganographie zu verbinden [ABS01]. Teilbilder wirken bei diesem Verfahren unauffälliger, da sie einen erkennbaren Inhalt anzeigen. Schließlich gibt es Erweiterungen, die die Kodierung farbige Bilder erlauben [VH97]. Für eine Übersicht über die Variationen der visuellen Kryptographie, siehe [Kl07]

3 Funktionalität und Anwendungsszenarien

3.1 Funktionsumfang

Die Applikation wird wie ein Malprogramm bedient und besitzt einen vergleichbaren Funktionsumfang. Der Benutzer wird mit einer Leinwand konfrontiert, auf der Zeichenoperationen ausführen kann. Auf der linken Seite befindet sich eine Palette mit Malwerkzeugen sowie ein Optionsmenü. In Letzterem werden, je nach Kontext, Liniendicke, Schriftart und -größe sowie andere Parameter angegeben. Erläutert werden die Malfunktionen mit ihren Parametern in Tabelle

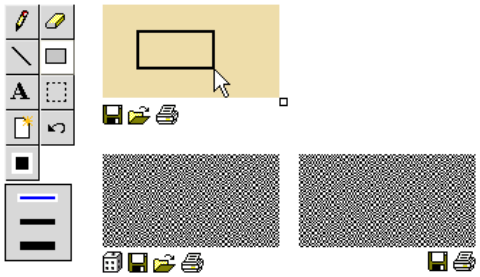

Abbildung 3: Screenshot der Applikation.

1. Unter Leinwand und den Teilbildern befinden sich Icons, die das Laden und Speichern ermöglichen. Sie werden in in Tabelle 2 behandelt.

Tabelle 1: Werkzeugpalette

	Zeichnet Freihhandlinien.
	Löscht Teile der Zeichenfläche.
	Zeichnet gerade Linien.
	Zeichnet Rechtecke.
A	Schreibt Text. Die Texteingabe geht von einer deutschen Tastaturbelegung aus.
	Selektiert einen Leinwandbereich und erlaubt, diesen zu verschieben. Befinden sich im selektierten Bereich unbenutzte Leinwandteile, können diese als transparente oder als deckende Flächen behandelt werden. Im Optionsmenü sind entsprechende Einträge vorhanden.
	Löscht die gesamte Leinwand.
	Macht die zueletzt ausgeführte Maloperation rückgängig. Die Anzahl der Schritte, die rückgängig gemacht werden können, ist unbegrenzt.
	Vertauscht Farbe des Vordergrundes mit der des Hintergrundes.

3.2 Nachrichtenübermittlung

Die visuelle Kryptographie bietet zwei Vorteile gegenüber anderen Verfahren: Sie ist kryptographisch sicher und einfach zu entschlüsseln. In folgendem Szenario könnten diese Vorteile ausschlaggebend sein: Ein Angestellter wird in ein Land geschickt, in dem keine sicheren Kommunikationskanäle existieren und keine Computer vorhanden sind. Der Arbeitgeber möchte ihm jedoch vertrauliche Informationen mitteilen können. Für diesen Zweck wird der Angestellte mit 10 Teilbildern ausgestattet. Der Arbeitgeber behält von den Bildern eine Kopie. Dadurch ist er in der Lage, Nachrichten zu kodieren, die mit einem der 10 Teil-

Tabelle 2: Lade und Speicherfunktionen

	Speichert ein Bild im GIF-Format. Im Fall der Leinwand wird die vertikale und horizontale Auflösung halbiert, um ein späteres wiedereinlesen zu ermöglichen.
	Lädt ein Bild im GIF- oder PNG-Format. Das Symbol unter der Leinwand dient dem Einbinden eines Bildes, das anschließend kodiert wreden soll. Das unter dem linken Geheimbild dagegen ermöglicht es, der Kodierung ein Teilbild vorzugeben.
	Druckt ein Bild in eine PDF-Datei. Ein Klick auf das Symbol unter dem rechte n Teilbild öffnet einen Auswahldialog. Dort kann der Benutzer angeben, beide Teilbilder in eine PDF-Datei zu drucken.
	Erzeugt neue Teilbilder. Eine Ausnahme wird für den Fall gemacht, dass de r Benutzer ein eigenes Teilbild eingebunden hat. In diesem Fall wird nur das rechte Geheimbild erneuert.
	Ändert die Größe der Leinwand. Der maximale Umfang beträgt 750x500 Pixel. Die Größe wurde so gewählt, dass ein Geheimbild vollständig einer PDF-Seit abgebildet wird.

bilder entschlüsselt werden können. Das bei der Kodierung entstehende Teilbild darf über einen unsicheren Kanal versendet werden wegen der kryptographischen Sicherheit des Verfahrens. Der Angestellte wiederum benötigt keinen Computer für die Entschlüsselung.
Ein Nachteil dieses Vorgehens besteht in der begrenzten Anzahl an Nachrichten, die der Arbeitgeber dem Angestellten senden kann. Jede Nachricht muss für ein unbenutztes Teilbild des Angestellten kodiert werden, ansonsten verliert das Verfahren an Sicherheit. In diesem Beispiel wäre die Anzahl der Nachrichten auf 10 begrenzt.

3.3 Authentifizierung

Eine weitere Anwendung der visuellen Kryptographie steht im Zusammenhang mit Online-Überweisungen. In ihrer regulären Form sind Online-Überweisungen anfällig für man-in-the-middle-Angriffe: Sendet der Benutzer eine Überweisung an die Bank, kann ein Dritter, der den Datenverkehr abfängt, die Kontonummer oder den Betrag ändern. Das Problem wird nicht umgangen, indem die Bank online eine Bestätigung an den Benutzer sendet. Denn auch diese kann gefälscht werden.
Visuellen Kryptographie ermöglicht es, die Bestätigung fälschungssicher zu machen. Dem Kunden werden auf Folien gedruckte Teilbilder per Post geschickt. Nach einer Überweisung sendet ihm die Bank ein Teilbild als Bestätigung zu. Der Benutzer legt eine seiner Folien darüber, um sie zu entschlüssen. Ein Dritter kann diese Art der Bestätigung nicht fälschen. Denn um ein Teilbild zu erzeugen,

das zusammen mit der Folie des Benutzers eine falsche Kontonummer anzeigt, muss er die Folie des Benutzers kennen.
Ein praktischer Nachteil des Verfahrens besteht in der Notwendigkeit, die Folien auf dem Monitor exakt zu platzieren. Eine segmentbasierte Variante der visuellen Kryptographie, wie sie in [B07] beschrieben wird, ist dafür potentiell besser geeignet.

4 Implementierung

Webseiten werden üblicherweise in HTML verfasst. Frühe Versionen dieser Sprache basierten auf einem statischen Seitenkonzept: Das Anzeigen eines neuen Inhalts erforderte das Laden einer neuen Seite. Es war nicht möglich, den Inhalt einer Seite zu ändern, sieht man von funktional begrenzten Konstrukten wie Textfeldern und Auswahldialogen ab. Technologien wie Java-Applets und Flash glichen diesen Nachteil aus. Sie konnten und können jedoch nur mithilfe eines Plugins genutzt werden.
HTML wurde schließlich um eine Skriptsprache (Javascript), Stildefinitionen (CSS) und eine dynamisch änderbare Dokumentstruktur (DOM) erweitert. Man fasst diese Technologien unter dem Begriff "dynamisches HTML" zusammen. Sie machten es u.a. möglich, Größe und Position von Bildern zu ändern sowie neue Texte und HTML-Elemente zu erzeugen.
Die Manipulationsmöglichkeiten beschränken sich jedoch auf Inhalte, die beim Laden der Seite übertragen oder durch den Benutzer generiert wurden. Das Nachladen server-generierter Inhalte ist mit dynamischen HTML nicht möglich.

4.1 AJAX

AJAX ("Asynchronous Javascript And XML ") erlaubt Server-Anfragen auch nach dem Laden einer Webseite. Dies geschieht ohne Reload der gesamten Webseite und setzt kein Plugin voraus. Aufgrund dieser Vorteile wurde AJAX zur Implementierung dieser Arbeit verwendet.
Der Kern von AJAX ist ein Javascriptobjekt namens XMLHTTPRequest, das HTTP-Requests erzeugen und Antworten des Servers speichern kann. Requests können asynchron ausgeführt werden. Die Applikation wird in diesem Fall nicht angehalten, bis der Server antwortet. Sattdessen werden die auf den Request folgenden Anweisungen unmittelbar bearbeitet. Sobald die Antwort eintrifft, springt die Applikation zu einer Funktion, die beim Erzeugen des Requests definiert wurde.
Die Antwort des Servers hat die Form einer Textdatei. Typischerweise enthält sie das Ergebnis einer Datenbankanfrage, HTML-Elemente oder Javascriptbefehle. Das "X"in AJAX spielt auf die Tatsache an, dass die Antworten häufig eine XML-Struktur aufweisen.
Eine Beispielanwendung der AJAX-Technologie ist die Startseite der deutschen Wikipedia. Tippt man einen Buchstaben in das Suchfeld, wird ein Request an

den Server gestellt. Dieser schickt eine Liste mit passenden Artikelnamen zurück, die dem Benutzer schließlich unterhalb des Suchfelds präsentiert werden.

4.2 Ablauf einer Zeichenoperation

Diese hier vorgestellte Applikation funktioniert auf eine ähnliche Weise. Mithilfe von Mausklicks und -bewegungen gibt der Benutzer Malbefehle. Ein Skript auf der Seite des Clients fängt diese Eingaben ab und erzeugt einen Server-Request. Der Server nimmt den Befehl entgegen, lädt eine Kopie der Leinwand und führt auf dieser den Befehl aus. Anschließend wird die Leinwand verschlüsselt. Danach werden alle Bilder auf den Client übertragen.

4.3 Optimierungen

Der Benutzer soll nicht mit Wartezeiten konfrontiert werden. Aufgrund der begrenzten Rechenkapazitäten des Servers und der möglicherweise schmalen Netzwerkverbindung können diese jedoch auftreten. Die nachfolgenden Optimierungen erlauben es den Zeichenoperationen, schneller einen sichtbaren Effekt auf die Leinwand zu haben.
Eine besteht darin, die Ausführung eines Zeichenbefehls von der Generierung der Teibilder zu trennen. Erst wenn der Server einen Malbefehl bearbeitet und dem Client eine neue Leinwand zugeschickt hat, wird die Kodierung ausgeführt.
Eine andere macht sich die in HTML spezifizierten DIV-Elemente zu Nutze. Es handelt sich hierbei um “Kästen” die mit Hintergrundfarbe versehen und pixelgenau dimensioniert sowie positioniert werden können. Mit ihnen sind Zeichenoperationen auch auf der Seite des Clients durchführbar. Diese Zeichenoperationen haben sofort einen sichtbaren Effekt, unabhängig von der Verbindungsqualität zum Server oder dessen Rechenkapazität. Jedoch ist die Anzahl der darstellbaren DIV-Elemente begrenzt. Wird ein Schwellwert überschritten, nimmt die Performance des Clients rapide ab. Aus diesem Grund werden DIV-Elemente wieder gelöscht. Dies kann unbemerkt geschehen, sobald eine neue Leinwand vom Server geladen wird.
Schließlich existiert eine Queue, in die Requests eingestellt werden können. Der Benutzer muss so nicht warten, bis ein Request abgearbeitet wurde, um einen neuen Zeichenbefehl zu erteilen.

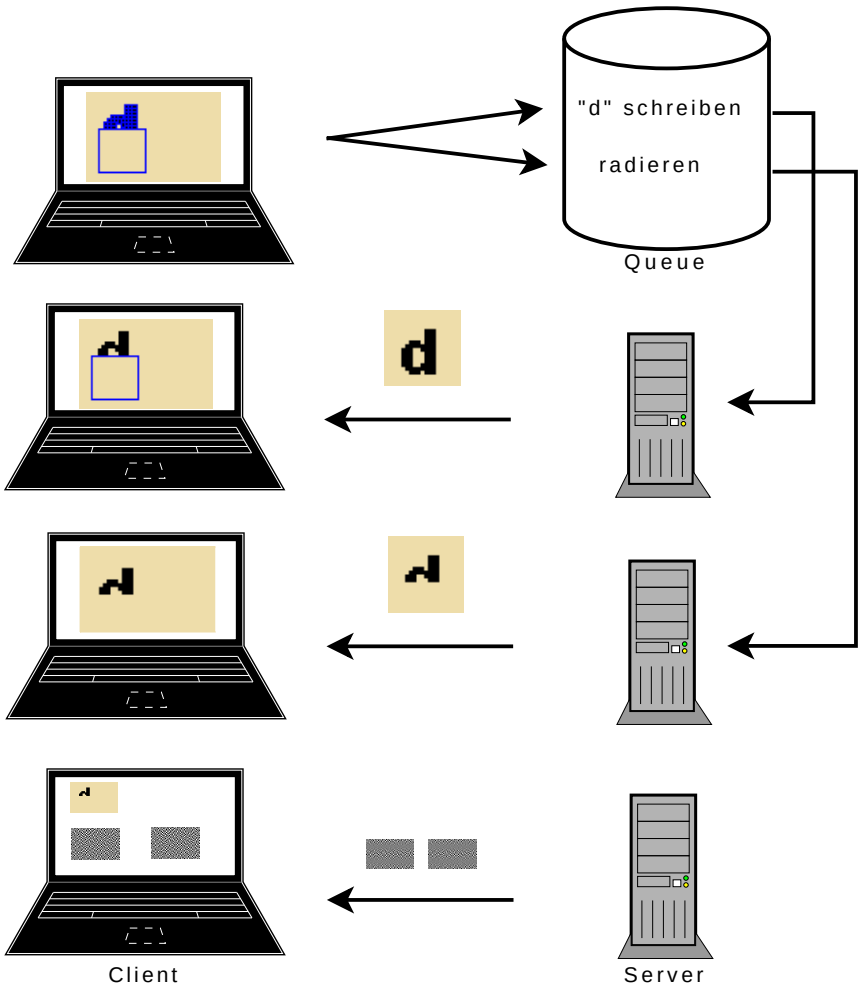

Abbildung 4: Ablauf eines Funktionsaufrufs. In der Ausgangssituation zeichnet der Benutzer ein “d”s und radiert einen Teil der Leinwand. Der Client führt diese Operationen mithilfe von DIV-Elementen aus, die hier blau hervorgehoben sind. Anschließend wird der Server informiert. Entsprechende Requests werden in eine Queue gestellt. Der Server arbeitet einen Befehl ab und schickt dem Client die neuen Leinwand-Bilder zu. Der Client löscht bei Erhalt unnötig gewordene DIV-Elemente. Schließlich erzeugt der Server neue Teilbilder, die auf den Client übertragen werden.

Literatur

[NS94] Moni Naor, Adi Shamir: Visual Cryptography. EUROCRYPT 1994: 1-12.

[Kl07] Andreas Klein: Visuelle Kryptographie. Springer Verlag, 2007.

[ABS01] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis und Douglas R. Stinson: Extended capabilities for visual cryptography. Theoretical Computer Science 2001: 143-161.

[VH97] Eric R. Verheul and Henk C.A. van Tilborg: Constructions and Properties of k out of n Visual Secret Sharing Schemes. Designs, Codes and Cryptography 1997 Volume 11: 179-196.

[B07] Bernd Borchert: Segment-based Visual Cryptography. Unveröffentlicht. Einsehbar unter www-fs.informatik.uni-tuebingen.de/~borchert/