

# Un attacco a basso costo su un Microsoft CAPTCHA

Jeff Yan  
School of Computing Science  
Newcastle University, UK  
Jeff.Yan@ncl.ac.uk

Ahmad Salah El Ahmad  
School of Computing Science  
Newcastle University, UK  
Ahmad.Salah-El-Ahmad@ncl.ac.uk

## ASTRATTO

CAPTCHA è ormai quasi una tecnologia di sicurezza standard. I CAPTCHA più diffusa sono *basato su testo schemi*, che in genere richiedono agli utenti di risolvere un compito di riconoscimento del testo. Lo stato dell'arte del design CAPTCHA suggerisce che tali sistemi basati su testo dovrebbero basarsi sulla resistenza di segmentazione per fornire garanzia di sicurezza, come il riconoscimento di carattere individuale dopo la segmentazione può essere risolto con un alto tasso di successo con i metodi standard come le reti neurali.

In questo articolo presentiamo le nuove tecniche di segmentazione carattere di valore generale per attaccare una serie di CAPTCHA di testo, compresi i regimi progettati e distribuiti da Microsoft, Yahoo e Google. In particolare, il CAPTCHA di Microsoft è stato distribuito a partire dal 2002 in molti dei loro servizi online tra cui Hotmail, MSN e Windows Live. Progettato per essere segmentazione- resistenti, questo schema è stato studiato e messo a punto dai suoi progettisti nel corso degli anni. Tuttavia, la nostra semplice attacco ha raggiunto un tasso di successo segmentazione superiore al 90% contro questo schema. Ci sono voluti in media ~ 80 ms per l'attacco completamente segmento fallo su un computer desktop normale. Di conseguenza, si stima che questo CAPTCHA potrebbe essere immediatamente interrotto da un bot malevolo con un totale (segmentazione e quindi il riconoscimento) tasso di successo di oltre il 60%. Anzi, L'obiettivo del progetto era che gli attacchi automatizzati non dovrebbero raggiungere un tasso di successo di superiore al 0,01%. Per la prima volta, questo documento dimostra che CAPTCHAs che sono accuratamente progettati per essere resistenti segmentazione-sono vulnerabili agli attacchi nuovi ma semplici.

## Categorie e descrittori Subject

D.4.6 Sicurezza e Tutela, H.1.2 utente / Machine Systems.

### Termini generali

Sicurezza, fattori umani.

### parole

CAPTCHA, la robustezza, l'attacco di segmentazione, l'usabilità, la sicurezza in Internet

## 1. INTRODUZIONE

Un CAPTCHA (completamente automatizzato Public Test di Turing to Tell Computers and Humans Apart) è un programma che genera e test i gradi che sono risolvibili umana, ma intende essere oltre le capacità dei programmi per computer attuali [1]. Questa tecnologia è ormai quasi un meccanismo di sicurezza standard per la difesa contro

Il permesso di fare copie digitali o cartacee di tutto o parte di questo lavoro per uso personale o in classe è concessa senza tassa a condizione che le copie non sono fatti o distribuito a scopo di lucro o vantaggio commerciale e che le copie portano questo avviso e la citazione completa sulla prima pagina. Per copiare altrimenti, o ripubblicare, scrivere un commento sul server o di ridistribuire alle liste, richiede la preventiva autorizzazione specifica e / o di una tassa.







CCS'08, 27-31 Ottobre 2008, Alexandria, Virginia, Stati Uniti d'America. Copyright  
2008 ACM 978-1-59593-810-7 / 08/10 ... \$ 5,00.

programmi bot indesiderati o dannosi su Internet, come quelli diffondere le email spazzatura e quelli afferrando migliaia di account di posta elettronica gratuito istantaneamente. E ha trovato ampia applicazione su numerosi siti web commerciali tra cui Google, Yahoo e MSN di Microsoft.

I CAPTCHA più usati sono i cosiddetti *basato su testo schemi*, che si basano su sofisticata distorsione delle immagini di testo volti a renderli irriconoscibili allo stato dell'arte dei metodi di riconoscimento di forme. La popolarità di tali sistemi è dovuta al fatto che essi hanno molti vantaggi [4], per esempio, essere intuitivo per gli utenti in tutto il mondo (la task utente eseguito essere solo il riconoscimento dei caratteri), avendo problemi di localizzazione poco (persone in diversi paesi tutti riconoscere i caratteri romani), e di un buon potenziale per fornire una protezione avanzata (ad esempio lo spazio di un attacco di forza bruta deve cercare può essere enorme, se il regime è stato progettato correttamente).

Un buon CAPTCHA deve essere non solo umana amichevole, ma anche abbastanza robusto per resistere ai programmi per elaboratore che gli aggressori scrivono di passare automaticamente i test CAPTCHA (o *sfalse*). (CAPTCHA è un ideale tema di ricerca nei giovani campo interdisciplinare della sicurezza utilizzabile, che ha acquisito sempre maggiore attenzione negli ultimi anni.)

Tabella 1. Tasso di riconoscimento dei caratteri singoli dal distorsioni differenti (tutti dati di questa tabella sono tratti da [6])

I personaggi sotto le distorsioni tipiche	Tasso di riconoscimento
	~ 100%
	96 +%
	100%
	98%
	~ 100%
	95 +%

Le prime ricerche ha suggerito che i computer sono molto bravi a riconoscere singoli caratteri, anche se questi personaggi sono molto [6] distorte. Tabella 1 mostra i caratteri sotto distorsioni tipiche, insieme con tassi di successo che una rete neurale può realizzare per

riconoscerli. Si è stabilito in [6] che se le posizioni dei personaggi sono conosciuti nelle immagini sfida generate da un CAPTCHA, quindi rompere questo schema è solo un problema di riconoscimento puro, che è un compito banale con tecniche di apprendimento automatico standard, come le reti neurali [12].

Tuttavia, quando la posizione di caratteri in una sfida CAPTCHA non è nota a priori (ad esempio nelle seguenti immagini prese da [4]), lo stato dell'arte (compresi apprendimento automatico) metodi non funzionano bene nel localizzare i caratteri, lasciate solo riconoscerli.



**Il problema di identificare posizioni dei caratteri nel giusto ordine, o *segmentazione*, è ancora un problema difficile nei campi come il riconoscimento della scrittura a mano e computer vision.** In generale, segmentazione è computazionalmente costoso, e spesso un problema combinatorio duro [4].

Lo stato dell'arte del design CAPTCHA suggerisce che la robustezza dei sistemi basati su testo dovrebbe contare sulla difficoltà di trovare in cui il personaggio è (segmentazione), piuttosto che quale personaggio è (il riconoscimento) [11, 3, 4, 5, 6]. **Cioè, tali CAPTCHA dovrebbe essere *la segmentazione-resistente*. In altre parole, *se la rottura di un CAPTCHA (basato su testo) può essere ridotto con successo ad un problema di riconoscimento dei caratteri individuali, allora questo schema è effettivamente rotto.***

In questo lavoro, riportiamo le nuove tecniche di segmentazione carattere di valore generale per attaccare una serie di CAPTCHA di testo, compresi i regimi progettati e distribuiti da Microsoft, Yahoo e Google.

In primo luogo, vi presentiamo un attacco di segmentazione romanzo su un alto profilo di Microsoft CAPTCHA. Progettato per essere segmentazione resistente, questo schema è stato uno sforzo di collaborazione di un team interdisciplinare di diverse competenze in Microsoft tra cui l'elaborazione dei documenti e la comprensione, apprendimento automatico, HCI e la sicurezza. In realtà, **il ampiamente accettato "resistenza segmentazione" Principio è stato sancito da questa squadra.**

Questo CAPTCHA è stato distribuito in molti dei servizi online di Microsoft, tra cui Hotmail, MSN e Windows Live per anni, con la sua prima versione utilizzata nel sistema di registrazione degli utenti di Hotmail nel 2002 [11]. Da allora, il regime è stato sottoposto a un miglioramento in termini di robustezza [3, 4, 6] e usabilità [4, 5]. Microsoft ha anche presentato tre domande di brevetto degli Stati Uniti per proteggere la tecnologia di base [8]. Chiaramente, questo sistema è stato progettato con cura.

Tuttavia, il nostro semplice e attacchi a basso costo ha raggiunto un tasso di successo segmentazione superiore al 90% sull'ultima versione di questo Microsoft CAPTCHA (come distribuito nell'estate del 2007) <sup>1</sup>. Per comodità, si farà riferimento a questo CAPTCHA come ***lo schema di MSN*** in questo documento. Con l'aiuto di questo attacco di segmentazione, si stima che il regime di MSN può essere rotto con un

---

**Il lavoro è stato fatto nell'estate del 2007. Abbiamo avvertito Microsoft la debolezza del loro CAPTCHA nel settembre 2007. In risposta alla loro richiesta, abbiamo tenuto questo attacco riservato fino al 10 aprile 2008. Per quanto a nostra conoscenza questo è il primo efficace attacco segmentazione del regime.**

complessiva (segmentazione e quindi il riconoscimento) tasso di successo di circa il 60%. Al contrario, il suo obiettivo di progettazione è stata che "script automatici non dovrebbero essere più successo di 1 su 10.000 (0,01%)" tentativi [4]. Inoltre, anche se lo schema di MSN è stato creduto di essere "estremamente difficile e costoso per i computer per risolvere" a causa della difficoltà di segmentazione che i suoi progettisti hanno introdotto [5], il nostro attacco completamente segmentato ogni sfida essenzialmente istantaneamente. **Al meglio delle nostre conoscenze, *questo per la prima volta dimostra che un CAPTCHA che è stato accuratamente progettato da professionisti seri come la segmentazione-resistenti è comunque vulnerabile agli attacchi romanzo, ma semplici.***

Successivamente, dimostriamo che il nostro attacco è applicabile anche ad altri CAPTCHA testo, inclusi gli schemi disegnati da Yahoo e Google. In particolare, una variante del nostro attacco ha raggiunto un alto tasso di segmentazione su un Yahoo CAPTCHA, che in teoria può portare all'attacco di maggior successo fino ad oggi sullo schema. La struttura dettagliata di questo lavoro è la seguente. Sezione 2 discute lavoro relativo. Sezione 3 rivede il sistema di MSN. Sezioni 4 e 5 dettaglio il nostro attacco e dei suoi risultati, rispettivamente. Sezione 6 discute l'applicabilità del nostro attacco. Si evidenzia una variante dell'attacco che abbiamo progettato per il CAPTCHA di Yahoo. Mostriamo anche che un componente dell'attacco è applicabile ad un CAPTCHA di Google e molteplici altri regimi. Sezione 7 discute rappresentativi meccanismi di resistenza "segmentazione" attuate fino ad oggi, scoprire altri esempi di vita reale di sicurezza e usabilità fallimenti in questo settore. Sezione 8 riassume questa carta e offrono conclusioni. Attaccando CAPTCHA ben progettato, implementato, impariamo come potrebbero fallire e potrebbe essere migliorata. Nel complesso, questo lavoro contribuisce al miglioramento immediato della sicurezza dei CAPTCHA che sono stati ampiamente distribuiti da Microsoft, Yahoo e Google, così come altri sistemi che presentano debolezze simili. Essa contribuisce anche a promuovere la nostra comprensione del progetto di CAPTCHA - la conoscenza collettiva in corso su questo argomento è molto limitato - per esempio, che i meccanismi di segmentazione resistenti concepite fino ad oggi sono deboli, ma che sembra essere sicuro contro gli attacchi attualmente disponibili. Attaccando CAPTCHA ben progettato, implementato, impariamo come potrebbero fallire e potrebbe essere migliorata. Nel complesso, questo lavoro contribuisce al miglioramento immediato della sicurezza dei CAPTCHA che sono stati ampiamente distribuiti da Microsoft, Yahoo e Google, così come altri sistemi che presentano debolezze simili. Essa contribuisce anche a promuovere la nostra comprensione del progetto di CAPTCHA - la conoscenza collettiva in corso su questo argomento è molto limitato - per esempio, che i meccanismi di segmentazione resistenti concepite fino ad oggi sono deboli, ma che sembra essere sicuro contro gli attacchi attualmente disponibili. Attaccando CAPTCHA ben progettato, implementato, impariamo come potrebbero fallire e potrebbe essere migliorata. Nel complesso, questo lavoro co

## 2. LAVORO CORRELATO

E 'stato riferito sul Feb 8, 2008 [17] che un aumento dello spam inviato dai conti di Windows Live è stata osservata, e un bot che potrebbe firmare conti Live Mail è stato analizzato da una società di sicurezza [18] per capire cosa ci fosse dietro questo fenomeno. Tuttavia, in questo caso riportato, la decodifica CAPTCHA non è stato fatto dal bot, ma ad un server remoto. Non è chiaro se ci fosse buon lavoro umano dietro la scena alimentazione manualmente risposte CAPTCHA. D'altra parte, anche se un attacco automatico è stato lanciato dal server, ad oggi, nessun dettaglio tecnico di questo attacco è stata rivelata a tutti. Inoltre, il tasso di successo osservato per il bot è stato solo circa il 30-35% [18]. La robustezza del CAPTCHA basato su testo è stato finora studiato principalmente solo nella visione di computer e di analisi dei documenti e le comunità di riconoscimento. Per esempio, Mori e Malik [9] hanno rotto EZ-Gimpy (92% di successo) e il Gimpy (33% di successo) CAPTCHAs con sofisticati algoritmi di riconoscimento dell'oggetto. Moy et al [10] hanno sviluppato tecniche di stima distorsione rompere EZ-Gimpy con un tasso di successo del 99% e 4- lettera Gimpy-r con un tasso di successo del 78%. Chellapilla e Simard [3] attaccato una serie di CAPTCHA visuali presi dal web

con la macchina di apprendimento algoritmi, il raggiungimento di un tasso di successo da 4,89% al 66,2%.

Il nostro primo lavoro [14] ha rotto una serie di CAPTCHA (compresi quelli ospitati a *Captchaservice.org*, un servizio web specializzato per la generazione CAPTCHA) con quasi il 100% di successo semplicemente contando il numero di pixel di ciascun carattere segmentato, anche se questi schemi erano resistenti alla miglior software OCR sul mercato. A differenza di altri lavori che si basava su sofisticati algoritmi di visione artificiale o di machine learning, questo studio ha utilizzato solo algoritmi di pattern recognition semplici ma sfruttato gli errori di progettazione fatali che sono stati scoperti in ogni schema. Questo è uno dei pochi lavori esaminando la robustezza del CAPTCHA dal punto di vista della sicurezza.

PWNtcha [7] è una pagina web eccellente che mira a "dimostrare l'inefficienza di molte implementazioni CAPTCHA". Si commenta brevemente i punti deboli di una dozzina di CAPTCHA semplici, che, secondo quanto affermato essere rotto con un successo che vanno dal 49% al 100%. Tuttavia, nessun dettaglio tecnico degli attacchi era a disposizione del pubblico. Molti altri CAPTCHA sono stati commentati in questo sito. Ad esempio, sia il regime MSN e Yahoo CAPTCHA che saranno discussi nel presente documento (cioè *Yahoo Schema 1* nella sezione 6.1) sono stati considerati da questo sito come "molto buono" e difficile da rompere.

sono stati proposti due algoritmi interessanti in [19] per amplificare il divario di competenze tra gli esseri umani e computer. Gli algoritmi potrebbero migliorare la sicurezza dei sistemi di CAPTCHA basati su testo, ma sono ortogonali a questo documento. (In questo lavoro, non discutiamo altri tipi di CAPTCHA, come quelli basati su immagini. Per coloro che sono interessati, una panoramica dei CAPTCHA basati su immagini può essere trovato in [19]).

Usabilità e robustezza sono due questioni fondamentali con CAPTCHA, e spesso interconnessione con l'altro. In [21], abbiamo esaminato problemi di usabilità che devono essere considerati e affrontati nella progettazione di CAPTCHA e discusso sottili implicazioni alcuni dei problemi possono avere sulla robustezza. Un'ultima nota: un sondaggio sulla ricerca CAPTCHA (compresa la progettazione della maggior parte dei sistemi di notevoli primi) può essere trovato in [13], e le limitazioni di difesa contro i bot con CAPTCHA (compresi gli attacchi a livello di protocollo) sono stati discussi in [15].

### 3. REGIME MSN

Fig 1 mostra alcuni sfide campione generati dal piano MSN CAPTCHA. Noi non abbiamo accesso al codice di base dello schema di MSN, così abbiamo raccolto dal sito web 100 campioni casuali di Microsoft che sono stati generati in tempo reale on-line su [16]. Studiando [4, 5] ed i campioni che abbiamo raccolto, abbiamo osservato che il regime di MSN (come schierato) ha le seguenti caratteristiche.



Fig 1. II MSN CAPTCHA: 4 sfide campione.

- Otto caratteri sono usati in ogni sfida;
- Solo lettere maiuscole e cifre vengono utilizzati.
- Primo piano (cioè sfida testo) è blu e sfondo scuro grigio chiaro.

- Orditura (sia locale che globale) viene utilizzato per la distorsione carattere.

Curvatura locale produce "piccole increspature, onde e deformazioni elastiche lungo i pixel del carattere", e lamine "algoritmi basati su caratteristiche che possono usare spessore caratteri o serif funzioni per rilevare e riconoscere caratteri" [6]. Caratteri nella prima e seconda fila di Tabella 1 sono ampiamente distorti dalla deformazione locale.

ordito globale genera carattere-livello, deformazioni elastiche a sventare algoritmi modello corrispondente per il rilevamento ed il riconoscimento di caratteri. Personaggi in terza e quarta fila della tabella 1 sono in gran parte distorte da deformazione globale.

- I seguenti archi casuali di spessori diversi sono usati come il principale anti-segmentazione.

o archi in primo piano spessore: Questi archi sono di primo piano colore. Il loro spessore possono essere uguali alle spesse parte dei caratteri. Essi non si intersecano direttamente con tutti i caratteri, così sono anche chiamati "archi non intersecanti".

o archi in primo piano sottili: Questi archi sono di primo piano colore. Anche se non sono tipicamente spessi come il tipo di archi sopra, lo spessore può essere lo stesso come le porzioni sottili di caratteri. Si intersecano con gli archi di spessore, personaggi o entrambi, e, quindi, anche chiamati "intersecano gli archi sottili".

o sfondo sottili archi: Questi archi sono sottili e di colore di sfondo. Hanno tagliato attraverso personaggi e rimuovere alcuni contenuti di carattere (pixel). Sia deformazione locale e globale è comunemente usato per distorsione nel CAPTCHA basati su testo. Molti schemi utilizzano texture di sfondo e maglie in primo piano e colori di sfondo, come il disordine per aumentare la robustezza. Però,

archi casuali di diverso spessori sono utilizzati come disordine nello schema di MSN. La logica è la seguente. Questi archi sono di per sé buoni candidati per i falsi caratteri. Il mix di archi e caratteri casuali confonderebbe lo stato dei metodi dell'arte di segmentazione, che fornisce una forte resistenza segmentazione [5].

### 4. UN ATTACCO SEGMENTAZIONE

Abbiamo sviluppato un attacco a basso costo che può efficace ed efficiente le sfide del segmento generato dal regime di MSN. Nello specifico, il nostro attacco raggiunga i seguenti:

- Identificare e rimuovere archi casuali
- Identificare tutte le posizioni dei caratteri nel giusto ordine; in altre parole, dividere ogni sfida in 8 segmenti ordinate, ciascuno contenente un singolo carattere.

Il nostro attacco è costruito su osservando e analizzando i campioni casuali 100 abbiamo raccolto - questo è un "campionario". L'efficacia di questo attacco è stato testato non solo sul set di prova, ma anche su un ampio set di prova di 500 campioni casuali - la progettazione dell'attacco utilizzata alcuna conoscenza preliminare su un campione che questo set. Questa metodologia segue la pratica comune nei campi quali la

computer vision e apprendimento automatico 2. ( Tutti i campioni sono stati raccolti nell'estate del 2007.)

Il nostro attacco coinvolge 6 passi consecutivi, ciascuno dei quali è descritti nelle seguenti sezioni.

4.1 Pre-processing

Abbiamo prima convertire una sfida ricca di colori in un'immagine in bianco e nero usando un metodo di soglia: pixel con intensità superiore ad un valore di soglia vengono convertiti in bianco, e quelli con una minore intensità di nero (vedere figura 2 (a) e (b)). La soglia è stata determinata analizzando manualmente campionario, e lo stesso valore è stato utilizzato per ogni immagine in entrambi i set di campioni e di test.

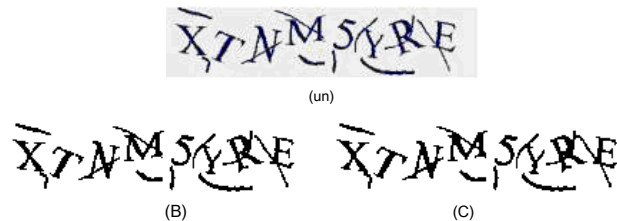


Figura 2. Pre-processing. (A) immagine originale, (b) immagine binarizzata, (C) dopo aver fissato i caratteri rotti.

(Questo esempio è tratto da [8], in cui la sua resistenza alla segmentazione è classificato da Microsoft come "duro", il livello più alto fra tutti gli esempi. Useremo questo esempio per illustrare l'intero processo del nostro attacco segmentazione in questo documento. ) la seconda fase di pre-elaborazione è fissare caratteri rotti: bassa sottili archi rimuovere alcuni contenuti carattere, e talvolta creano una crepa in caratteri (ad esempio, il secondo carattere "T" in figura 2 (a) è rotto a causa di questo ragionare). Questa fase ha un duplice scopo: i) per mantenere un carattere come una singola entità e di conseguenza migliorare i nostri metodi di segmentazione di follow-up, e ii) impedire piccole porzioni di caratteri venga rimosso ad arco di rumore in seguito.

Abbiamo osservato che sfondo sottili archi sono in genere 1-2 pixel di larghezza dopo binarizzazione, e il seguente metodo semplice funziona bene per identificare e risolvere i caratteri rotti causati da tali archi. (1) Trova pixel che sono di colore di sfondo e hanno lasciato e

vicini di destra con colore di primo piano (vedi fig 3 (a)). (2) Trova i pixel che sono del colore di sfondo e hanno superiore e

vicini inferiori con colore di primo piano (vedi Figura 3 (b)). (3) Convertire pixel sopra individuate al colore di primo piano.

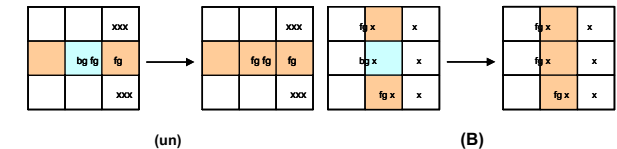


Fig 3. Collegamento gap 1-pixel ( 'x' rappresenta un pixel che è sia in primo piano o il colore di sfondo).

Questo metodo consente il collegamento qualsiasi gap 1-pixel che soddisfa le condizioni illustrate nella figura 3. Il suo effetto è illustrato nella figura 2 (c): alcuni pixel mancanti per carattere 'T' sono recuperati. Un effetto collaterale di questo metodo è che si potrebbe introdurre ulteriori pixel di primo piano che

• In uno studio correlato [10] pubblicato CVPR'04, la conferenza premier computer vision, la dimensione del set campione era 564 e la dimensione del test di set 736.

collegare componenti che vengono inizialmente scollegati. Ad esempio, nella Figura 2 (c), un sottile arco interseca con 'R' è ora collegato con un altro arco interseca con 'E'. Ma questo inconveniente si è dimostrata un problema trascurabile nel nostro studio - che non sarebbe il caso se abbiamo scelto di collegare tutte le lacune di due pixel.

4.2 Segmentazione verticale

Un metodo di segmentazione verticale viene applicato per segmentare una sfida in verticale in più pezzi, ciascuno dei quali può contenere uno o più caratteri. Il processo di segmentazione verticale inizia mappando l'immagine a un istogramma che rappresenta il numero di pixel di primo piano per colonna nell'immagine. Quindi, le linee di segmentazione verticali separano l'immagine in pezzi tagliando attraverso le colonne che non hanno pixel di primo piano affatto. Fig 4 mostra che tale istogramma verticale segmentazione taglia una sfida in due pezzi.

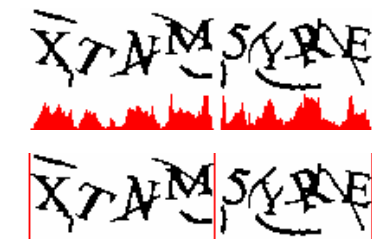


Fig 4. Composizione verticale Segmentazione

Tipicamente, questo metodo verticale non solo realizza segmentazione parziale, ma contribuisce anche alla nostra strategia divide et impera, che è fondamentale per il successo del nostro attacco.

4.3 Colore di riempimento la segmentazione

In questa fase, una "segmentazione riempimento colore (CFS)" algoritmo viene applicato a ogni chunk segmentato nel passaggio precedente. L'idea di base di questo algoritmo è rilevare ogni componente collegato, che noi chiamiamo oggetto, in un pezzo. Un oggetto può essere un arco, carattere, archi collegati, o caratteri connessi. L'algoritmo funziona come segue. In primo luogo, rilevare un pixel di primo piano, e quindi tracciare tutti i suoi vicini di primo piano finché tutti i pixel di questa componente collegato vengono mossi - cioè, viene rilevato un oggetto. Successivamente, l'algoritmo individua un pixel di primo piano fuori area dell'oggetto rilevato (s), ed inizia un altro processo attraversamento per identificare un oggetto successivo. Questo processo continua fino a quando tutti gli oggetti del pezzo si trovano. Questo metodo è efficace come usare un colore distinto per inondare ciascun componente collegato, si chiama perciò segmentazione "riempimento colore". Alla fine, il numero di colori usati per riempire un pezzo è il numero di oggetti nel pezzo. Con il nostro metodo CFS, come mostrato in figura 5 (a),

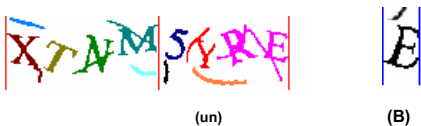


Fig 5. Colore segmentazione riempimento

Spesso, una sfida è diviso in quattro o cinque pezzi di segmentazione verticale. Vale la pena ricordare che questo fase di riempimento colore viene applicato ad ogni pezzo, piuttosto che solo i pezzi più ampi che, probabilmente, contengono più di un oggetto. La ragione è semplicemente che pezzi sottili possono anche contenere più di un oggetto (vedi figura 5 (b)), e occorre individuare tutti gli oggetti in ogni blocco e

monitorare il numero di oggetti per la rimozione dell'arco follow-up e altri passaggi.

CFS contribuisce ad un'ulteriore segmentazione rilevando oggetti che non possono essere segmentato con il metodo verticale, e fornisce il numero di oggetti in ogni blocco. Come verrà discusso più avanti, CFS contribuisce anche ad ulteriori fasi come la rimozione arco.

#### 4.4 rimozione arco di spessore

archi spessi, se presenti, vengono rilevati e rimossi dopo il processo di riempimento colore sopra.

**Caratteristiche di archi.** Per motivi di usabilità, di spessore

archi in primo piano non si intersecano con i caratteri sfida, a meno che non siano collegati indirettamente attraverso un arco sottile (archi sottili si intersecano con caratteri) o sono costretti a connettersi con gli altri a causa l'inconveniente introdotto dal metodo di fissazione caratteri rotti nella Sezione 4.1. Abbiamo anche osservato che gli archi spessore hanno le seguenti caratteristiche, che permettono di identificare e rimuovere automaticamente.

- **numero di pixel.** Spesso, un arco di spessore è relativamente piccolo numero di pixel (cioè, il numero di pixel di primo piano nel arco).
- **Posizione.** archi spessi sono situati in prossimità o addirittura si intersecano con il bordo dell'immagine, qualcosa che raramente si verifica con i caratteri validi a meno che non siano collegati al arco di spessore.
- **Forma.** archi spessi non contengono cerchi. Personaggi come A, B, D, P, Q, 4, 6, 8 e 9, tutti contengono uno o più cerchi.
- **Interazione tra la forma e la posizione.** La posizione di archi spesse e le loro forme geometriche sono in qualche modo correlati. Ad esempio, gli archi spesse situate all'inizio e alla fine di una sfida sono tipicamente alto ma stretto (cioè, il rapporto dell'altezza sulla larghezza è grande); archi spessore nella parte centrale di una sfida tendono ad essere ampia ma breve (cioè il rapporto tra la larghezza su altezza è grande).

**algoritmo di rimozione Arc.** Il nostro algoritmo è in gran parte basato sulle osservazioni di cui sopra, e comprende i seguenti passaggi.

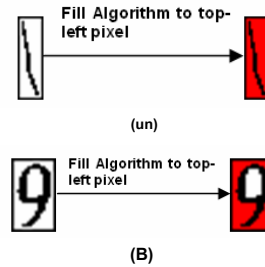
##### 1) rilevazione Cerchio, che rileva se un oggetto contiene un cerchio.

Se un oggetto contiene un cerchio, sappiamo che non è sicuramente un arco, e tutti gli altri metodi di rimozione arco può essere saltato. Il metodo di rilevazione cerchio funziona nel modo seguente.

- Disegnare un riquadro attorno a un oggetto, in modo che la casella di delimitazione non tocchi alcuna parte dell'oggetto.
- Applicare l'algoritmo di riempimento di colore per il pixel in alto a sinistra, vale a dire, inondare tutti i pixel dello sfondo che sono collegati al pixel in alto a sinistra, con un colore che è diverso dal primo piano e sfondo

- Eseguire la scansione del riquadro di delimitazione per pixel del colore di sfondo. Se viene trovato un tale pixel, allora viene rilevato un cerchio. Altrimenti, viene rilevato alcun cerchio. Figura 6 illustra due casi di esempio. In figura 6 (a), non v'è alcun pixel del colore di sfondo originale una volta applicato l'algoritmo di riempimento. Cioè, siamo sicuri che questo oggetto non contiene alcun cerchi. Al contrario, l'algoritmo di riempimento non può liberarsi di tutti i pixel del colore di sfondo originale in figura 6 (b). Pertanto, rilevando questi pixel, l'algoritmo è sicuro che un cerchio esiste in questo oggetto. (Per migliorare l'efficienza

dell'algoritmo di riempimento, il gap minimo tra l'oggetto e il riquadro di delimitazione è solo un pixel in entrambi i casi.)



rilevamento Figura 6. Circle: esempi

Poi, usiamo i seguenti 3 passaggi per rilevare e rimuovere archi spessi come segue. Al termine di ogni fase, l'istogramma dell'immagine viene aggiornata.

##### 2) La scansione di tutti gli oggetti che non contengono cerchi per discriminante

**Caratteristiche** (Gli altri oggetti vengono ignorati). Tale discriminazione è in gran parte circa il controllo numero di pixel. Se un oggetto ha un numero di pixel inferiore o uguale a 50, viene rimosso come un arco. (Abbiamo osservato che tipicamente un personaggio ha un numero di pixel di dimensioni superiori a 50). Quando questo passaggio è stato applicato alla sfida in figura 5 (a), un arco nel 2<sup>do</sup> chunk è stato rimosso a causa della sua piccola numero di pixel (vedi Figura 7).



rimozione Fig 7. Arc - controllo funzione discriminativa: un arco il secondo pezzo viene rimosso.

##### 3) controllo posizione relativa. Questo passaggio esamina la relativa

posizione degli oggetti in un pezzo, ed è applicata a tutti i blocchi che contengono più di un oggetto (si noti che i caratteri collegati vengono considerati come un unico oggetto). L'idea alla base di questo passaggio è che le posizioni relative degli oggetti possono dire archi e personaggi reali a parte. Ad esempio, tipicamente caratteri sono più vicini alla linea di base (cioè la centrale orizzontale di un pezzo), mentre gli archi sono più vicini ai bordi dell'immagine superiore o inferiore.

Inoltre, personaggi sono orizzontalmente giustapposti, ma mai in verticale. Una volta che questo passaggio è completato, l'istogramma viene aggiornato. Come mostrato in figura 8, quando questo metodo è stato applicato alla sfida in figura 7, ulteriori archi sono stati rimossi. Nel frattempo, l'istogramma è stato aggiornato, e l'immagine è stata ulteriormente segmentata.



rimozione Fig 8. Arc - posizione relativa verifica: Ulteriori archi sono stati rimossi e istogramma è stato aggiornato.

Il controllo posizione relativa si è dimostrato il più efficace nella rimozione di archi nel nostro attacco. Un elenco incompleto di matrici di posizioni relative tipico è illustrato con esempi reali in Tabella 2.

Tabella 2. matrici di posizioni relative tipici schemi

posizione relativa			Decisione
disposizione	Descrizione	Esempio	
O1 O2 O3	<i>Tre oggetti in un pezzo:</i> due oggetti più o meno allineati lungo la linea di base, la 3a oggetto nella presenza di loro		O3 è arc
O3 O1 O2	<i>Tre oggetti in un pezzo:</i> due oggetti più o meno allinearsi lungo la linea di base, il terzo oggetto sopra o di loro		O3 è arc
O0 O1 O2 O3	<i>Quattro oggetti in un pezzo:</i> Tre oggetti più o meno allineati lungo la linea di base, il 4 ° oggetto sotto qualsiasi di essi		O3 è arc
O1 O2 O3 O4	<i>Quattro oggetti in un pezzo:</i> Due oggetti più o meno allineati lungo la linea di base, il 3 ° e 4 ° oggetti sotto e sopra uno di essi, rispettivamente,		O1 e O4 sono archi
O1 O2	<i>Due oggetti in un pezzo:</i> verticalmente giustapposte		O O1 o O2 *

\* Prima applicare il risultato di rilevamento cerchio ottenuto prima: se solo uno degli oggetti contengono un cerchio, quindi l'oggetto senza un cerchio viene rimosso come un arco. Se questo non funziona, allora l'oggetto che viene meno allineata con la linea di base viene rimosso come un arco.

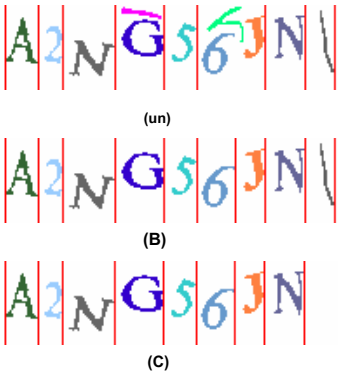
4) Rilevamento di archi rimanenti. I passaggi di cui sopra non lo fanno

necessariamente identificare tutti gli archi in un'immagine. Ciò che viene fatto in questa fase è la seguente. In primo luogo, contare il numero di oggetti rimanenti nell'immagine (archi individuati sono già rimossi e quindi non conteggiati). Se questo numero è maggiore di 8, allora c'è almeno un arco inosservato nell'immagine. Un sorprendente osservazione su questi arco inosservato (s) è che spesso sono il primo o l'ultimo oggetto nell'immagine corrente. Un metodo ad hoc lavora per la maggior parte dei casi è sufficiente selezionare le prime e le ultime oggetti con le seguenti regole:

- Se solo uno di essi contiene un cerchio, l'oggetto senza un cerchio viene rimosso come un arco.

- Se nessuno di essi contiene un cerchio, quindi l'oggetto con un numero di pixel più piccolo viene rimosso. Questo processo si ripete fino a quando l'immagine non ha esattamente 8 oggetti rimanenti.

Un altro esempio che illustra l'intero processo di rimozione arco è nella figura 9, dove (a) era un'immagine segmentata da segmentazioni verticali e CFS. La funzione discriminante checking omesso di rilevare qualsiasi arco, ma posizione relativa checking rilevato un arco sia il 4 ° e 6 blocchi. Figura 9 (b) è il risultato dopo quei due archi vengono rimossi e l'istogramma è stato aggiornato. Poi, sfuggita al rilevamento archi catturato l'ultimo oggetto come un arco. L'immagine finale al termine del processo di rimozione dell'arco è la figura 9 (c).



rimozione Fig 9. Arc: un altro esempio.

4.5 Individuazione personaggi collegati

Dopo aver rimosso archi, un passo immediato è quello di individuare eventuali caratteri connessi, quali la segmentazione riempimento verticale o colore non è riuscita a segmento. Tra  $n$  oggetti in uscita dalla fase precedente, se  $n < 8$ , allora almeno uno degli oggetti contiene due o più caratteri, questi caratteri sono collegati (tipicamente da archi di intersezione sottili). Questo passaggio stima quanti caratteri sono collegati e li individua. Le seguenti caratteristiche di progettazione e di attuazione del programma MSN contribuiscono ad essere in grado di valutare quali oggetti contengono il numero di caratteri collegati.

- Lunghezza fissa: ogni sfida utilizza 8 caratteri.
- caratteri collegata in oggetto sono orizzontalmente ma mai verticalmente giustapposte. Pertanto, un oggetto contenente due o più collegati caratteri è tipicamente più larga altri oggetti.
- In media un pezzo segmentato - per definizione, un pezzo non può essere ulteriormente suddiviso dal metodo verticale ma può dal metodo CFS - contiene più caratteri se il pezzo è maggiore di 35 pixel. (Questa larghezza è stato misurato dopo il seguente processo di normalizzazione è stato applicato al pezzo: La linea segmentazione sinistra viene regolata ad attraversare la sinistra pixel più primo piano nel chunk verticalmente e analogamente per la linea segmentazione destra) Secondo il numero di blocchi, la larghezza di ogni blocco, e il numero di oggetti in ciascun blocco, possiamo immaginare con un alto tasso di successo che chunk / oggetto contiene caratteri collegati e il numero di questi caratteri (o in altre parole, indovinare quanti caratteri esistono in ogni blocco ).



Usiamo due esempi per mostrare come funziona il nostro algoritmo. L'istogramma dell'immagine in figura 8 indica che contiene quattro pezzi. Dato che ci sono esattamente 8 caratteri in questi pezzi, sappiamo che ci sono le seguenti cinque possibilità esclusiva per la distribuzione di tutti i personaggi tra i pezzi 3:

(un) Ci sono quattro blocchi, ciascuno con due caratteri. (B) Un pezzo ha tre personaggi e ci sono due blocchi aggiuntivi ciascuno avente due caratteri. (C) Un pezzo ha quattro personaggi e altri due personaggi. (D) Ci sono due pezzi ciascuno con tre personaggi. (E) Un pezzo ha cinque caratteri. Poiché la 2<sup>nd</sup>, 3<sup>rd</sup> e 4<sup>esimo</sup> blocchi dell'immagine erano tutti maggiore di 35 pixel, l'algoritmo determina che vi sono almeno tre blocchi ciascuno avente più di un carattere. Di conseguenza le opzioni (c), (d) e (e) sono esclusi - nessuna delle opzioni permetterebbe più di due pezzi che hanno più di un carattere. L'algoritmo conosce anche dal algoritmo di CFS che il 2<sup>nd</sup>

blocco contiene tre oggetti, e quindi l'opzione (a) è anche caduto. Questo lascia unica opzione (b); così l'algoritmo identifica che il 2<sup>nd</sup> blocco contiene esattamente tre personaggi e la 3<sup>rd</sup> e 4<sup>esimo</sup> pezzi contiene due caratteri ciascuno.



Fig 10. "ravvicinamento" per l'individuazione di caratteri collegati

Il secondo esempio (vedi figura 10) è più sottile. L'istogramma su questo indica che contiene 5 pezzi. Dato che ci sono esattamente 8 caratteri in questi pezzi, sappiamo che ci sono tre possibilità esclusivi per la distribuzione di tutti i personaggi tra i pezzi:

(un) Uno dei pezzi contiene 4 caratteri (b) Un pezzo ha tre personaggi e altri due personaggi. (C) Ci sono tre pezzi ciascuno con due caratteri. Poiché i pezzi 3° e 4° dell'immagine erano maggiore di 35 pixel, l'algoritmo determina che almeno 2 doppie esiste e conseguentemente opzione (a) è esclusa. Dal momento che ci sono stati solo due di tali pezzi più ampi, opzione (c) è anche caduto. Questo lascia unica opzione (b).

Per determinare quale blocco contiene una tripla e che contiene un letto, l'algoritmo confronta la larghezza ed il numero di oggetti in entrambi i blocchi. L'algoritmo scoprire che il 3<sup>rd</sup> chunk "MG" è il pezzo più largo, ma sa anche dal algoritmo di CFS che il 4<sup>esimo</sup> chunk "28G" contiene 3 oggetti, questo lascia solo un massimo di 2 oggetti che possono esistere nel 3<sup>rd</sup> pezzo; così l'algoritmo identifica che il 3<sup>rd</sup> blocco contiene due caratteri collegati.

È possibile ottenere gli stessi risultati senza utilizzare il numero di blocchi ma basandosi seguito sul numero di oggetti. Tuttavia, questo metodo alternativo richiede tenere traccia non solo la posizione di ciascun oggetto nell'immagine, ma anche la posizione rispetto al

<sup>3</sup> Nel caso generale, è anche banale enumerare tutte le possibilità per la distribuzione di 8 caratteri attraverso qualsiasi proposta  $c$  ( $c$  è un numero intero tra 1 e 8) blocchi. D'altra parte, nei nostri esperimenti, gli scenari in cui  $c = 1, 2$  o  $3$  sono mai avvenuto.

i suoi vicini, che renderebbe molto più complicato per implementare l'algoritmo.

#### 4.6 Segmentazione personaggi collegati

Il passo precedente ha identificato un oggetto (s) contenente caratteri collegati, così come il numero di questi caratteri, indicato con  $c$ , contenuto in ciascun oggetto. Abbiamo osservato che spesso, un metodo semplice "anche tagliare" lavora per segmentare i personaggi collegati in un oggetto come segue.

1) Elaborare la larghezza dell'oggetto identificando sua sinistra più e più a destra pixel;

2) Verticalmente dividere l'oggetto in  $c$  parti della stessa

larghezza, ogni parte essendo un segmento corretta. Ad esempio, è stato determinato che l'ultimo oggetto in figura 8 e la 3<sup>rd</sup> oggetto in figura 10 contiene due caratteri collegati. Per questi oggetti, quali l'algoritmo non è quello di dividere equamente in due segmenti, ciascuno un personaggio. Figura 11 mostra i finalizzate 8 segmenti per entrambe le sfide.



Fig 11. immagini completamente segmentate

#### 5. RISULTATI

**Tasso di successo.** Il nostro attacco di segmentazione ha raggiunto un tasso di successo del 91% sul set del campione. Cioè, 91 su 100 sfide sono stati segmentati in modo corretto. Per verificare se è stato abbastanza generica, abbiamo fatto il nostro attacco su una serie di test di 500 sfide casuali - il nostro programma ha avuto alcuna conoscenza preliminare su un campione che questo set. Il nostro attacco ha raggiunto un tasso di successo del 92% sul set di test (la distribuzione di campioni nel test set privilegia leggermente il nostro algoritmo). Per entrambi gli insiemi di campioni e di test, il tasso di successo è stata stabilita manualmente.

Abbiamo analizzato tutti i casi di fallimento del nostro attacco di segmentazione sia nel campione e insiemi di test, e abbiamo scoperto che tre tipi di errore si è verificato nel modo seguente.

- La mancata rimozione di arco: alcuni archi di spessore sono stati rilevati.
- Fallimento di individuare personaggi collegati. Un caso tipico è: quando un singolo carattere (es 'W') era molto più ampia di due caratteri collegati, il primo, piuttosto che quest'ultimo, potrebbe essere identificato come quello contenente caratteri connessi. D'altra parte, quando gli archi spessore non sono stati rilevati ma trattati come caratteri validi, potrebbero anche causare nostro algoritmo di sicuro per rilevare caratteri collegati.
- Fallimento della "anche tagliare". E 'sorprendente che questo semplice metodo non sempre funziona a personaggi segmento collegato.

Abbiamo anche confrontato la percentuale di ciascun tipo di errore sia nel campione e set di test. I modelli di guasto di entrambi i gruppi sono simili. I dettagli della nostra analisi dei guasti sono in [22].

**velocità di attacco.** Abbiamo implementato il nostro attacco in Java (poco sforzo è stato speso per ottimizzare la fase di esecuzione di codice), e testato su un computer desktop con un 1,86 GHz Intel Core 2 CPU e 2 GB di RAM. L'attacco è stato eseguito dieci volte sul set sia il campione e di test, e la velocità media è stata scattata (vedere Tabella 3). I dati riportati nella tabella mostrano che il nostro attacco è molto efficiente: in media, ci vogliono

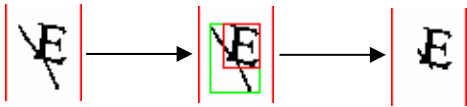
solo un po' più di 80 ms completamente segmento una sfida in entrambi i set.

Tabella 3. Velocità Attacco

Velocità (Ms / sfida) Media Max Min			
Fascicolo di prova	82.8	91,4	81,4
test set	84.2	95,5	82,8

**Implicazioni.** Stato dell'arte di machine learning può raggiungere un tasso di successo di almeno il 95% per il riconoscimento di singoli caratteri nello schema di MSN, dopo che sono stati segmentati [5, 6]. Tuttavia, questo tasso è una stima prudente per il riconoscimento dei caratteri nei campioni che abbiamo raccolto per questo studio, per i seguenti motivi.

- In primo luogo, abbiamo controllato tutti i campioni nel nostro set test dopo abbiamo misurato il tasso di successo del nostro attacco, e abbiamo trovato che, sebbene gli stessi tipi di tecniche di distorsione sono stati applicati ai caratteri nei nostri campioni e quelli elencati nella Tabella 1, i primi erano molto meno distorta rispetto al secondo. La stessa osservazione applicato anche al set di campionamento.
- In secondo luogo, controllando manualmente tutti i campioni che sono stati correttamente segmentata per il nostro attacco, abbiamo osservato nessun artefatto che verrebbero introdotte da qualsiasi fase dell'attacco di interferire con il passaggio a titolo definitivo.
- In terzo luogo, abbiamo metodi semplici per sbarazzarsi di alcune porzioni di "intersecanti archi sottili" in ogni personaggio segmentato in modo che questi personaggi sono ancora meno distorta e di conseguenza più facile da essere riconosciuto da tecniche di apprendimento automatico standard. Ad esempio, uno dei nostri metodi è quello di indovinare la zona del vero carattere all'interno di un oggetto controllando la densità di pixel di primo piano per l'oggetto. Come illustrato in Figura 12 (dove l'esempio è preso dalla ultimo segmento in Figura 11 (a)), la maggior parte delle colonne e righe all'interno della scatola rossa hanno un conteggio di pixel maggiore di un valore di soglia (3 in questo caso), mentre per parti esterne di questa scatola, la maggior parte delle colonne e delle righe hanno un numero di pixel inferiore, che è nella gamma degli spessori degli archi di intersezione sottili. Pertanto, porzioni di tali archi sono giustamente riconosciuti e rimossi come distorsione.



rimozione arco Fig 12. Thin utilizzando dell'imballo base pixel-densità scatola di stima.

Come tale, il nostro attacco segmentazione suggerisce che il regime di MSN può essere rotto con almeno una complessiva (segmentazione e riconoscimento) tasso di successo del 61% ( $\approx 0,92 * 0,95 \wedge 8$ ).

## 6. APPLICABILITÀ

Il nostro attacco contro il regime di MSN è applicabile ad altri CAPTCHA. In questa sezione, discuteremo alcuni casi.

### 6.1 Yahoo CAPTCHA

Abbiamo applicato con successo una variante del nostro attacco ad un CAPTCHA che è stato distribuito da Yahoo a loro siti web a livello mondiale fino a poco tempo - l'ultimo giorno che abbiamo osservato questo schema era in attivo

utilizzare (sul sito di Yahoo in Cina) è stato l'8 marzo, 2008. Il nostro attacco ha raggiunto un tasso di segmentazione di circa il 77% su questo CAPTCHA. Di conseguenza, si stima che questo schema potrebbe essere rotto con un totale (segmentazione e quindi il riconoscimento) tasso di successo di circa il 60% ( $\approx 77 * .95 \wedge 5$ ; la durata media testo in questo schema è 5). Cioè, in teoria, il nostro lavoro può portare all'attacco di maggior successo fino ad oggi sullo schema. **Messa in allarme, Yahoo ha cessato di utilizzare questo CAPTCHA.**

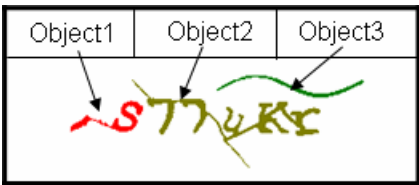
Fig 13 mostra alcuni esempi sfide generate da questa Yahoo CAPTCHA, che chiamiamo **Yahoo Schema 1**. Analizzando 100 campioni casuali, abbiamo osservato che l'uso di archi intersecanti era il meccanismo di resistenza segmentazione principale in questo schema, e gli archi potrebbe avere lo stesso spessore come alcune porzioni di caratteri validi.



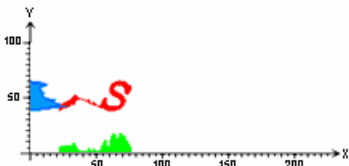
Fig 13. Yahoo Schema 1: Esempio sfide.

Il nostro attacco a questo schema funziona come segue. Dopo binarizing un'immagine, abbiamo segmentato in un insieme di componenti collegati (cioè oggetti) applicando il metodo CFS - questo metodo non solo realizza segmentazione parziale, ma contribuisce anche alla nostra strategia divide-impera.

Poi, per ciascun oggetto, si usa un metodo, che si estende dalla segmentazione verticale in sezione 4.2, per rilevare e rimuovere archi. Questo metodo è un'estensione significativa nostro lavoro sul regime MSN, e la sua tecnica chiave è la seguente analisi istogramma. In primo luogo, mappiamo ciascun oggetto di due istogrammi una che rappresenta il numero di pixel di primo piano per colonna, e l'altra che rappresentano il numero di pixel di primo piano per riga in oggetto. Li X e istogrammi Y- chiamiamo, poiché sono creati come se l'oggetto è proiettata rispettivamente gli assi X e l'asse Y. Figura 14 (b) mostra X- (di colore verde) e istogrammi Y- (in colore blu) per ciascuno dei tre oggetti identificati nella Figura 14 (a) con il metodo CFS.

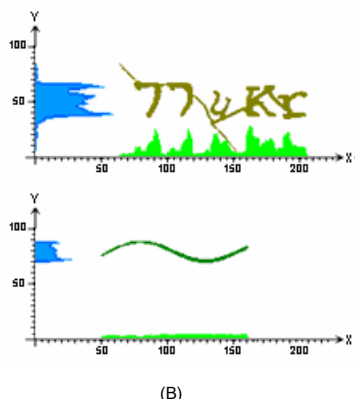


(un)



Una squadra di sicurezza russo ha affermato che essi hanno rotto lo stesso schema con un successo di circa il 35% [20]. Nessun dettaglio tecnico del loro attacco è stato a disposizione del pubblico, tuttavia.





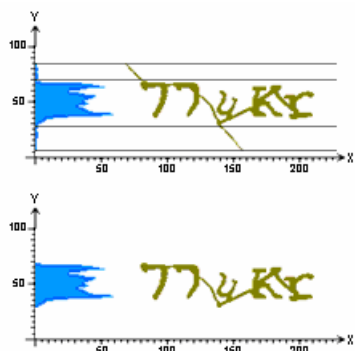
**Fig 14. (a) Il risultato di CFS; (B) X e Y per istogrammi ogni oggetto identificato.**

Quindi, l'algoritmo rimozione arco è principalmente una sequenza ordinata di analisi dell'istogramma, e funziona come segue.

In primo luogo, l'algoritmo verifica il più alto valore di picco di istogrammi di ciascun oggetto. Se il valore di picco di una sua X o Y-istogramma è troppo piccolo, allora l'oggetto è troppo piatto o sottile per essere un carattere valido, e viene rimosso come un arco. Quando è stato applicato questo passaggio, il terzo oggetto in Figura 14 (a) è stato correttamente rimosso come un arco, ma gli altri due stati.

In secondo luogo, l'algoritmo esamina Y- istogramma di ciascun oggetto rimanente per identificare *righe bassa densità*, che hanno solo un piccolo numero di pixel. Quando un numero sufficiente di righe (almeno 4 nei nostri esperimenti) sono consecutivi, tipicamente costituiscono una regione che ha una bassa densità di pixel di primo piano. Tale regione indica in genere che queste righe contengono solo (porzioni) archi, e può essere rimossa.

Come mostrato nella figura 15, questo passaggio correttamente rimosso porzioni di archi in entrambe le aree superiore ed inferiore del secondo oggetto in Figura 14 (a), anche se ha avuto alcun effetto sul primo oggetto.

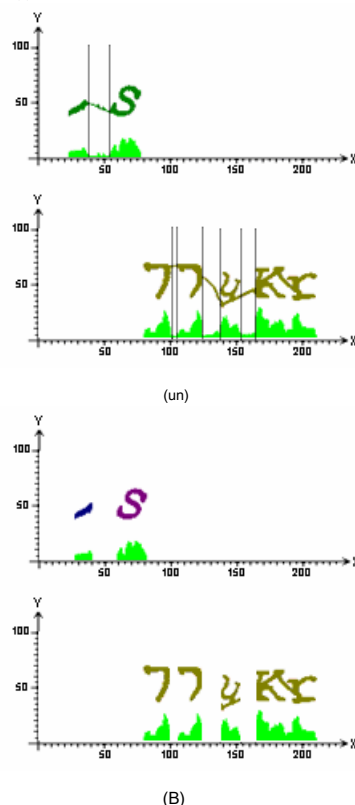


**rimozione Fig 15. Arc: (a) righe bassa densità sono identificate, e (B) dopo la fase 2.**

Se qualsiasi arco viene rimosso da un oggetto, X-istogramma dell'oggetto deve essere aggiornato alla fine di questa fase (per le sake sia di efficienza e la precisione di ulteriore rimozione arco).

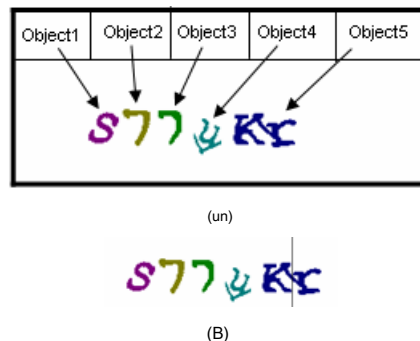
Come terzo passo, l'algoritmo esamina X-istogramma di un oggetto per identificare *colonne bassa densità*. Quando un numero sufficiente di tali colonne sono consecutivi, costituiscono una regione che ha una bassa densità di pixel di primo piano. Tale regione indica in genere che queste colonne sono (porzioni) archi che possono essere rimossi in modo sicuro.

Come mostrato in figura 16, questo passaggio rimosso correttamente alcune porzioni orizzontali di archi in entrambi gli oggetti.



**rimozione Fig 16. Arc: (a) le colonne a bassa densità sono identificate, e (b) dopo il punto 3.**

Infine, ripulire. Alcune piccole porzioni di archi possono ancora rimanere dopo le fasi di cui sopra, ad esempio il primo oggetto in Fig 16 (b). Tuttavia, queste porzioni tendono ad avere un numero di pixel molto più piccolo di qualsiasi carattere valido, e quindi sono facili da identificare e rimuovere. Figura 17 (a) mostra l'immagine sfida dopo l'intero processo di rimozione arco. A quanto pare, il nostro algoritmo non solo rimuove archi standalone, ma contribuisce anche alla segmentazione rimuovendo porzioni di archi che collegano diversi personaggi.



**Fig 17. (a) Dopo la rimozione arco, e (b) una sfida segmentato.**

Dopo la rimozione arco, usiamo un metodo che è molto simile alla sezione 4.5 per localizzare caratteri rimanenti connessa e stimare il numero di tali caratteri. Infine usiamo lo stesso metodo "anche tagliare" come nella sezione 4.6 di segmentare loro.

Ad esempio, per l'immagine nella Figura 17 (a), l'algoritmo determina che il più probabile era che Object5 aveva due caratteri collegati per le sue dimensioni, e quindi l'oggetto è stato uniformemente segmentato a due parti. Figura 17 (b) mostra il risultato della segmentazione finale, che è corretto.

Un'analisi dettagliata fallimento per il nostro attacco su schema Yahoo 1 è disponibile in [2].

## 6.2 Google CAPTCHA

Abbiamo anche testato un CAPTCHA che viene distribuito da Google per proteggere i loro servizi on-line (vedi Fig 18) con il nostro attacco alla schema di MSN. Abbiamo correttamente segmentato 12 di 100 campioni casuali che abbiamo raccolto, portando ad un tasso di successo del 12%. Ciò potrebbe portare ad un tasso di successo complessivo del 8,7% ( $\approx .12 * .95 \wedge 6,25$ , la lunghezza media testo in questo schema è 6,25). Tuttavia, il successo di segmentazione è un contributo esclusivamente con il metodo del CFS. Al momento di preparare la versione camera-ready del presente lavoro, sembra che Google ha fissato questa vulnerabilità.



Fig 18. Il Google CAPTCHA: sfide campione.

## 6.3 altri CAPTCHA

Vale la pena di notare che sia gli schemi di Yahoo e Google abbiamo discusso in precedenza sono stati progettati per essere resistenti segmentazione. Per CAPTCHA che non seguono il principio della resistenza di segmentazione, sarebbe banale per il metodo CFS di segmentare in modo corretto. Ad esempio, il metodo CFS sarebbe un **modo più efficiente ed efficace di attaccare Captchaservice.org** schemi che si era spezzato nel nostro precedente lavoro [14].

## 7. Sulla segmentazione RESISTENZA

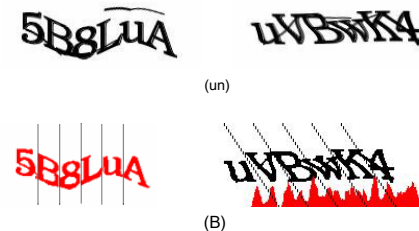
I CAPTCHA di Microsoft, Yahoo e Google di cui sopra rappresentano tre stili principali di meccanismi di resistenza segmentazione attuate fino ad oggi, che sono riassunte come segue.

- Lo stile Microsoft: archi casuali come falsi caratteri.
- Lo stile di Yahoo: linee di collegamento ad angolo casuali.
- Lo stile di Google: affollamento personaggi insieme. Applicando il nostro romanzo attacco segmentazione, abbiamo identificato che questi meccanismi, come attualmente implementati, hanno falle di sicurezza. Tuttavia, noi non pretendiamo che il principio di resistenza segmentazione è ribaltata. Ad esempio, è fattibile per la difesa contro il nostro attacco sul sistema di Google, eliminando spazi tra caratteri adiacenti ad attaccare quest'ultimo insieme - questo sarebbe del tutto sconfiggere il nostro attacco. (Tuttavia, questo potrebbe peggiorare un problema di usabilità che, come discusso in seguito, esiste già nell'attuale implementazione del sistema, se non si prendono precauzioni). Ci sono anche metodi semplici per migliorare il sistema di MSN, ad esempio:

- Adottando il metodo di "affollamento personaggi insieme", ad esempio, lasciando che i personaggi si toccano o si sovrappongono tra loro.
- Rendendo più difficile dire caratteri e archi a parte (ad esempio giustapponendo caratteri e archi in ogni direzione).
- Utilizzando in modo casuale varie larghezze per i caratteri potrebbe anche confondere alcuni parti del nostro attacco.

Sebbene non vi sia ancora alcuna prova tecnica conclusiva, il metodo di "affollamento personaggi insieme", se attuato correttamente, sembra avere più merito rispetto ad altri metodi nella fornitura di resistenza segmentazione. Ad esempio, come discusso in precedenza, può essere applicato per migliorare sia gli schemi di Microsoft e Google.

Probabilmente motivata dalla stessa osservazione, Yahoo srotolato il loro nuovo CAPTCHA nel marzo 2008. Come mostrato in figura 19 (a), i testi sfida in questo schema sono più compatti rispetto a prima, e dei personaggi sono di solito collegati - che o contatto con l'altro, o sono collegati da linee che si intersecano casuali. Usiamo questo ultimo schema di Yahoo come l'ultimo esempio cautelativo in questo documento per mostrare come un principio apparentemente sana può andare storto nella pratica.







di Figura 19. Yahoo ultimo piano (a) Esempio sfide; (B) immagini segmentate

Abbiamo scoperto un certo numero di difetti elementari, ma fatali in questo ultimo schema di Yahoo. Per esempio, sarebbe difficile o addirittura impossibile per un attacco automatizzato di segmentare una sfida, se il numero di caratteri nella sfida è sconosciuto. A differenza del sistema di MSN, nuova CAPTCHA del Yahoo utilizza una variegata lunghezza del testo, che è una caratteristica buon design. Tuttavia, abbiamo osservato **che il numero di caratteri ( $n$ ) in una sfida può essere stimata con un alto tasso di successo** misurando la larghezza del testo nella sfida. Inoltre, questo schema è vulnerabile a uno una versione semplificata del nostro attacco sullo schema Yahoo precedente, o un nuovo attacco "segmentazione angolare" che i segmenti di una sfida correttamente con linee angolate. Il primo esempio in Figura 19 (b) mostra un caso estremo, in cui una sfida è **vulnerabile al primo attacco: un "taglio uniforme" lavorato dopo  $n$  è stato stimato**. Il secondo esempio in Figura 19 (b) mostra che una sfida è stata correttamente segmentato per linee angolate. Utilizzando due di questi algoritmi di segmentazione semplici con regole associate per identificare quale algoritmo da utilizzare, abbiamo raggiunto un tasso di successo di segmentazione di circa il 33,4% sulle ultime schema di Yahoo. Di conseguenza, si stima che questo schema può essere rotto con un tasso di successo complessivo (segmentazione e riconoscimento) del 25,9% ( $\approx 0,334 * 0,95 \wedge 5$ ; la durata media testo in questo schema è 5). La nostra analisi di sicurezza dettagliata di questo schema di Yahoo è discusso in [2]. Abbiamo informato Yahoo questo attacco, così come l'attacco descritto nella Sezione 6.1. In risposta alla loro richiesta, abbiamo mantenuto il nostro lavoro riservate per consentire loro il tempo di correggere le vulnerabilità.

D'altra parte, mentre il metodo "affollamento personaggi insieme", se attuato correttamente, sembra fornire una maggiore sicurezza, si può introdurre una lacuna usabilità che è stato a lungo ignorato, vale a dire un nuovo tipo di caratteri confusi. Ad esempio, in alcune distorsioni nel sistema di Google, "vv" assomiglia a "w"; "Cl" assomiglia "d"; "Nn" assomiglia "m"; "Rn" assomiglia "m"; "Rm" assomiglia "nn"; "Cm" assomiglia "an", e così via (vedi tabella 4 per un paio di esempi). Nel 2007, abbiamo osservato che il 6% delle sfide generate dal sistema di Google conteneva tali caratteri, e sarebbe a malapena essere utilizzabile per gli utenti umani, o per lo meno








creerebbero confusione in modo che gli utenti non potevano essere sicuro di quello che le risposte giuste dovrebbero essere.

Tabella 4. caratteri Confondere in Google CAPTCHA

Immagine	caratteri Confondere
	la parte centrale è 'd' o collegato "CL"?
	Un altro caso di "cl" o "D" confusione.
	la parte iniziale è 'm' o collegato 'm'? Un vero e
	proprio mai di testa: è la prima parte 'm' o "m", al centro parte "inv" o "nw"?

Un simile problema di usabilità esiste anche nel più recente sistema di Yahoo, che adotta il metodo "affollamento caratteri insieme" (vedi Tabella 5 per alcuni esempi). Abbiamo osservato che circa il 10% delle sfide generate dal presente sistema contiene tali caratteri confusi, e quindi sarebbe umano irrisolvibile o almeno causare confusione. Tuttavia, questo problema è stato raramente osservato in Yahoo Schema 1.

Tabella 5. caratteri Confondere l'immagine più recente Yahoo schema sfida

in	Risposta
	yKKV5y o yKKT5y?
	SFrsFe o sFrsEe?
	HZKA8S o HKA8S?
	CRAR o crdr?
	znAzWg o zn4zwG?
	Non ho idea di che cosa il secondo carattere è
	6LmuF o 6LmuF?

Dato il gran numero di sfide inutilizzabili osservati, si consiglia che qualsiasi sistema che implementa questa "crowding caratteri insieme" meccanismo trattare coppie di caratteri confusi con particolare attenzione durante distorsione loro. Inoltre, per CAPTCHA, come l'ultimo schema di Yahoo, sembra che non si utilizza linee che si intersecano a tutti migliorerebbe ulteriormente l'usabilità del programma, senza sacrificare la sicurezza.

8. SOMMARIO E CONCLUSIONE

Per la prima volta, abbiamo dimostrato che, sebbene MSN CAPTCHA della Microsoft basa intenzionalmente la sua robustezza su resistenza segmentazione, è vulnerabile ad un semplice, a basso costo attacco segmentazione. Il nostro attacco ha raggiunto una segmentazione

tasso di successo del 92%, e questo implica che il regime di MSN può essere rotto con un totale (segmentazione e quindi il riconoscimento) tasso di successo di oltre il 60%. Pertanto, il nostro lavoro dimostra che il regime di MSN fornisce solo un falso senso di sicurezza. Testato dai suoi progettisti, lo schema di MSN è resistente agli attacchi arte segmentazione precedente. Tuttavia, per la prima volta, abbiamo usato un metodo di riempimento colore per segmentare caratteri in un CAPTCHA. Insieme con l'analisi tradizionale verticale istogramma, questo metodo ha dimostrato potente. Abbiamo anche trovato che è facile dire automaticamente archi casuali (che sono stati utilizzati come falsi caratteri nello schema di confondere gli attacchi automatizzati) da caratteri validi esaminando caratteristiche come numero di pixel, forme, luoghi, posizioni relative, e distanze al basale. Abbiamo anche progettato un nuovo metodo per l'individuazione caratteri connesse e stimare il numero di tali caratteri. L'attacco al sistema di MSN è stato testato anche su altri CAPTCHA. In particolare, una variante dell'attacco ha raggiunto un alto tasso di segmentazione su un CAPTCHA che è stato ampiamente distribuito da Yahoo fino all'inizio di quest'anno. Inoltre, una componente dell'attacco, cioè la segmentazione CFS, è applicabile alla Google CAPTCHA e molteplici altri regimi. I CAPTCHA di Microsoft, Yahoo e Google abbiamo analizzato rappresentati tre principali meccanismi di resistenza segmentazione attuate fino ad oggi. Mentre i meccanismi utilizzati nella MSN e Yahoo (Schema 1) CAPTCHA sono stati rotti dai nostri attacchi, sembra che il "affollamento personaggi insieme" meccanismo sostenuto dal CAPTCHA di Google potrebbe fornire una migliore protezione contro gli attacchi attualmente disponibili. L'attacco al sistema di MSN è stato testato anche su altri CAPTCHA. In particolare, una variante dell'attacco ha raggiunto un alto tasso di segmentazione su un CAPTCHA che è stato ampiamente distribuito da Yahoo fino all'inizio di quest'anno. Inoltre, una componente dell'attacco, cioè la segmentazione CFS, è applicabile alla Google CAPTCHA e molteplici altri regimi. I CAPTCHA di Microsoft, Yahoo e Google abbiamo analizzato rappresentati tre principali meccanismi di resistenza segmentazione attuate fino ad oggi. Mentre i meccanismi utilizzati nella MSN e Yahoo (Schema 1) CAPTCHA sono stati rotti dai nostri attacchi, sembra che il "affollamento personaggi insieme" meccanismo sostenuto dal CAPTCHA di Google potrebbe fornire una migliore protezione contro gli attacchi attualmente disponibili. L'attacco al sistema di MSN è stato testato anche su altri CAPTCHA. In particolare, una variante dell'attacco ha raggiunto un alto tasso di segmentazione su un CAPTCHA che è sta

Tuttavia, questo meccanismo non è stato senza preoccupazioni. Per la prima volta, abbiamo individuato alcuni difetti di questo meccanismo come attuata in Google e gli ultimi sistemi di Yahoo. Inoltre, abbiamo identificato un problema di usabilità a lungo ignorato introdotta da questo sempre più popolare meccanismo resistente segmentazione. Abbiamo anche discusso le contromisure per affrontare questi sicurezza e usabilità preoccupazioni. Ci aspettiamo che con tutti i miglioramenti apprese dai fallimenti precedenti, il "affollamento personaggi insieme" meccanismo diventerà più robusto e facile da usare. Nel complesso, tutti questi contribuiscono a promuovere attuale comprensione della progettazione di CAPTCHA migliori, in particolare la progettazione e realizzazione di meccanismi di resistenza segmentazione. Per concludere questo lavoro, abbiamo la seguente. disegno CAPTCHA è un argomento interdisciplinare in cui le competenze di più domini gioca un ruolo importante. Come dimostrato in questo lavoro, competenze di ingegneria di sicurezza ed esperienza, in particolare le capacità di pensiero contraddittorio (cioè identificazione di ciò che può andare storto), possono dare un contributo unico e significativo al miglioramento della robustezza del CAPTCHA, ma non erano sul posto quando Microsoft , Yahoo o Google stavano progettando i loro schemi.

Un'altra lezione importante è che, anche se la resistenza segmentazione è un sano principio, il diavolo è nei dettagli. Le tecniche che hanno riferito in questo documento, in particolare quelli usati sul MSN e due CAPTCHA Yahoo, dimostrazione di metodi per valutare la forza dei meccanismi di resistenza segmentazione. La relativamente ampia applicabilità del nostro attacco sul sistema di MSN è incoraggiante. Tuttavia, dubitiamo che ci sia un attacco di segmentazione universale che si applica a tutto il testo

CAPTCHA, visto che esistono centinaia di varianti di progetto [19]. Invece, un'aspettativa più realistica è quella di creare una serie di strumenti (vale a dire un insieme di algoritmi e attacchi, idealmente organizzate in modo componibile) per valutare la forza della CAPTCHA - questo è il nostro lavoro in corso.

Progettare CAPTCHA che presentano sia una buona robustezza e usabilità è molto più difficile che potrebbe sembrare. L'attuale comprensione collettiva di questo argomento è ancora nella sua infanzia. Di evolvere la progettazione di CAPTCHA, un giovane ma importante argomento, da un'arte in una scienza richiede ancora un notevole studio. La nostra esperienza suggerisce che CAPTCHA passerà attraverso lo stesso processo di sviluppo evolutivo come la crittografia, watermarking digitale e simili, con un processo iterativo, in cui gli attacchi riusciti portare allo sviluppo di sistemi più robusti.

## 9. RINGRAZIAMENTI

Siamo grati a Brian Randell per la correzione delle bozze una prima versione di questa carta e molti commenti utili. Commenti e suggerimenti da Philippe Golle e revisori anonimi anche contribuito a migliorare questo documento.

## 10. RIFERIMENTI

[1] L von Ahn, M Blum e J Langford. "Gli esseri umani e Telling Computer A parte automaticamente", CACM, V47, No2, 2004. [2] J Yan e AS El Ahmad. "E 'manodopera a basso costo dietro la scena"

- attacchi automatizzati a basso costo su Yahoo CAPTCHA", Scuola di Informatica Relazione Scienze Tecniche, Università di Newcastle, in Inghilterra, 2008. [3] K e P Chellapilla Simard, "Utilizzo di Machine Learning per

Rompere visiva interazione umana prove", informazioni neurali Sistemi di elaborazione (PIN), MIT Press, 2004. [4] K Chellapilla, K Larson, P e M Simard Czerwinski,

"Costruire segmentazione basata interazione umano-friendly prove", 2<sup>nd</sup> Int'l Workshop sulle Prove interazione umana, Springer-Verlag, LNCS 3517, 2005. [5] K Chellapilla, K Larson, P e M Simard Czerwinski,

"Progettare amichevoli prove di interazione uomo umani", ACM CHI'05, 2005. [6] K Chellapilla, K Larson, P Simard, M Czerwinski,

"I computer battere gli umani a riconoscimento singolo carattere in lettura a base di interazione umana Prove", 2a Conferenza su e-mail e Anti-Spam (CEAS), 2005. [7] Sam Hocevar. PWNtcha - captcha sito web decoder,

<http://sam.zoy.org/pwntcha/>, si accede gennaio 2008. [8] Microsoft Corporation. "Interazione uomo Proof (HIP) - Tecnica e Market Overview", 2006. Disponibile all'indirizzo [http://download.microsoft.com/.../Human\\_Interaction\\_Proof\\_Technical\\_Overview.doc](http://download.microsoft.com/.../Human_Interaction_Proof_Technical_Overview.doc). Accessed gennaio 2008. [9] G e J Mori Malik.

"Riconoscendo oggetti in contraddittorio disordine: rompere un CAPTCHA visivo", IEEE Conference on Computer Vision e Pattern Recognition (CVPR) 2003.

[10] G Moy, N Jones, C Harkless e R Potter. "Distorsione tecniche di stima a risolvere CAPTCHA visiva", IEEE CVPR, 2004.

[11] P Simard, R Szeliski, J Benaloh, J Couvreur e Calinov,

"Utilizzo di riconoscimento dei caratteri e la segmentazione di raccontare i computer da esseri umani", Conferenza internazionale sulla analisi dei documenti e riconoscimento (ICDAR), 2003. [12] P Simard, D Steinkraus, J Platt.

"Best Practice per

Convolutionale Reti Neurali applicati all'analisi documento visivo", Conferenza internazionale sulla analisi dei documenti e riconoscimento (ICDAR), IEEE Computer Society, Los Alamitos, pp.958-962, 2003. [13] C Papa e K Kaur. "Is It umani o computer? difendere

E-Commerce con CAPTCHA", IEEE Esperto informatico, marzo 2005, pp. 43-49 [14] J Yan e AS El Ahmad. "CAPTCHA visivi Rottura

con Naive Pattern Recognition Algorithms", in *Proc. del 23<sup>rd</sup> Conferenza annuale Computer Security Applications (ACSAC'07)*. FL, Stati Uniti d'America, dicembre 2007. IEEE Computer Society. pp 279-291. [15] J Yan. "Bot, Cyborg e Automated Test di Turing", il

Quattordicesima Workshop internazionale sulla Sicurezza Protocolli, Cambridge, UK, Marzo 2006. Disponibile anche in

<http://www.cs.ncl.ac.uk/research/pubs/trs/papers/970.pdf>. [16] <https://signup.live.com/hmnewuser.aspx?>

ci & revipc = CN & ts = 3970181 & sh = WsBO & HM = 1 & ru = http% 3A% 2F% 2fmail.live.com% 2F% 3fnewuser% 3dyes & rx = http% 3a% 2f% 2fget.live.com% 2fmail% 2foverview & rollrs = 04 & lic = 1 [17 ] Dan Goodin, "crack Automated Automated per Windows

Live captcha si scatena", The Register, 8 febbraio, 2008.

[http://www.theregister.co.uk/2008/02/08/microsoft\\_captcha\\_Buster/](http://www.theregister.co.uk/2008/02/08/microsoft_captcha_Buster/) [18] Websense Security Labs, "Streamlined anti-CAPTCHA

operazioni di spammer su Microsoft Windows Live Mail", Feb 6, 2008.

<http://securitylabs.websense.com/content/Blogs/2907.aspx> [19] J Elson, JR Douceur, J Howell e J Saul. "Asirra: un CAPTCHA che sfrutta interesse allineati manuale dell'immagine categorizzazione". ACM CCS'07. [20] "Yahoo! CAPTCHA è rotto", disponibile all'indirizzo [http://Network-](http://Network-broken.html)

[broken.html](http://Network-broken.html) security-research.blogspot.com/2008/01/yahoo-captcha-is-. [21] J Yan e AS El Ahmad. "Usabilità dei CAPTCHA - Oppure,

problemi di usabilità nella progettazione CAPTCHA", il quarto simposio sul utilizzabile privacy e sicurezza, Pittsburgh, USA, luglio 2008. [22] J Yan e AS El Ahmad. "Un attacco a basso costo su un

Microsoft CAPTCHA", Scuola di Informatica Relazione Scienze Tecniche, Università di Newcastle, in Inghilterra, 2008.