

مقاوم‌سازی و بهبود کپچای متنی با استفاده از جدول تلاقی حروف و دایره رنگها

علی شمسایی^۱، محسن سرداری^۲

^۱ دانشگاه آزاد اسلامی یزد

^۲ دانشگاه میبد

چکیده

در این مقاله به ارائه روشی جهت مقاوم‌سازی کپچای متنی می‌پردازیم. در این روش ما به تعریف نقاط تلاقی می‌پردازیم و سپس جدولی متشکل از نقاط تلاقی در تمامی حروف تشکیل می‌دهیم حال با بهره‌گیری از جدول تلاقی و نقاط تلاقی می‌توانیم به صورت اصولی درهم‌سازی حروف را انجام دهیم. در این روش با استفاده از جدول تلاقی مانع از غیرقابل تشخیص شدن کپچای به وجود آمده می‌شویم به صورتی که کپچای پایانی توسط کاربر کاملاً خوانا باشد. هدف از انجام این پژوهش جلوگیری از شکسته شدن کپچای متن توسط الگوریتم‌های حاضر است و می‌خواهیم با استفاده از درهم‌سازی حروف در کپچا مانع از تقسیم‌بندی و تشخیص حروف توسط الگوریتم‌های OCR شویم. علاوه بر این باید درهم‌سازی به گونه‌ای انجام شود که خوانایی کلمه حفظ شود و باعث ناخوانا شدن کپچا نشود و کاربران به راحتی بتوانند کلمه کپچا را خوانده و وارد کنند.

واژه‌های کلیدی: کپچای متنی، مقاوم‌سازی کپچا، نقاط تلاقی حروف، امنیت کپچا.

۱. مقدمه

با پیشرفته‌تر شدن دنیای فناوری و سرعت انتقال اطلاعات، گسترش مجامع مجازی و وب سایت‌ها، حملات به سایت‌ها از طریق ربات‌ها و اسکریپت‌های از پیش نوشته‌شده رو به افزایش است و هرکس هرروزه سعی می‌کنند تا با استفاده از روش‌های مختلف به اطلاعات مهم وب‌سایت‌ها و پایگاه‌های اطلاعاتی دسترسی پیدا کنند. در نمونه‌های کوچک‌تر آن‌ها می‌توانند با نوشتن برنامه‌ای ساده پسوندهای امنیتی کاربران را حدس زده و از اطلاعات محرمانه آن‌ها استفاده کنند، پس از این‌رو نیاز به روشی برای مقابله با این مشکل است.

Completely Automated Public Turing Test To Tell Computers And Humans Apart یا به عبارتی تست عمومی کاملاً خودکار تورینگ برای تشخیص انسان از کامپیوتر^۱ یک سامانه‌ی امنیتی و روند ارزیابی است که برای جلوگیری از برخی حمله‌های خرابکارانه‌ی ربات‌های اینترنتی به کار می‌رود. این روند می‌تواند مشخص کند که مراجعه‌کنندگان به یک وب‌سایت و یا سایر خدمات آنلاین انسان هستند یا کامپیوتر است. بدین منظور برنامه‌ی کپچا آزمون‌هایی را تولید می‌کند که تنها انسان‌ها قادر به پاسخ‌گویی به آن‌ها باشند و کامپیوترها و نرم‌افزارهای فعلی نمی‌توانند پاسخ درستی به این آزمون‌ها بدهند، پس هر کاربری که آن را به‌درستی حل کند، انسان فرض می‌شود (پندی و لوت^۱، ۲۰۱۴).

با پیشرفت شاخه‌های علم کامپیوتر از جمله پردازش تصویر راهکارهای متعددی جهت کشف متن کپچا از پس‌زمینه و افکت‌های موجود در آن ارائه شد که تا درصد قابل قبولی به‌درستی متن کپچا را حدس می‌زد (چنداوال و سپکال^۲، ۲۰۱۰). از آن‌پس کپچا‌های مختلفی از قبیل کپچای تصویر و صدا ارائه شد تا امنیت آن را بالا ببرد اما همان‌طور که گفته شد با گذر زمان آن‌ها نیز تا حدودی آسیب‌پذیر معرفی شدند (چو و تیگر^۳). از آنجاکه پیشرفت علم در عرصه‌های پردازش تصویر و هوش مصنوعی منجر به شکستن راحت‌تر کپچاهای متن و تصاویر شده پس باید راهکاری را ارائه کرد که با پیشرفته‌تر کردن کپچاها مانع از شکستن آن‌ها شویم و این موضوع را هم باید در نظر بگیریم که استفاده نادرست از کپچا وعدم رعایت اصول آن می‌تواند موجب کاهش ارتباط کاربران با سایت گردد.

همه‌روزه به دلیل ضعف کپچا حملات متعددی به سرورها و سایت‌ها می‌شود که از این‌رو با پیاده‌سازی این روش می‌توان تا حد زیادی ضعف کپچا را برطرف کرد و از آن حملات جلوگیری کرد.

در صورت ادامه این مشکل سرویس‌دهندگان متقبل هزینه‌های سنگینی می‌شوند، در تحقیقات به عمل‌آمده کپچاهای موجود توسط الگوریتم‌های تشخیص متن و پردازش تصویر به‌راحتی قابل حل است و شکسته می‌شود که راه ورود هرکس و ربات‌ها را به سایت‌ها و سرورها باز می‌کند. در تحقیقات گذشته نیز سعی در پیچیده کردن کپچا شده اما به‌مرورزمان باز روش‌هایی جهت شکستن آن‌ها نیز ارائه شده است که تا درصد بالایی در حدود ۹۴٪ عملی بوده است (استاروستنکو و همکاران^۴، ۲۰۱۵).

۲. کارهای گذشته

در این مقاله بر روی کپچای متن تمرکز می‌کنیم زیرا هم از نظر پیاده‌سازی و هم از نظر هزینه به‌صرفه است. از OCR برای شکستن کپچای تصویر مبتنی بر متن استفاده می‌شود که به‌طور کلی آن را در سه مرحله انجام می‌دهد: ۱- پیش‌پردازش که تصویر

¹ Pandey and Lothe

² Chandavale and Sapkal

³ Chew and Tygar

⁴ Starostenko et al.

را برای پردازش آماده می‌سازد، ۲- تقسیم‌بندی که تصویر را قطعه‌بندی کرده به‌طوری‌که هر ناحیه دارای یک کاراکتر باشد، ۳- شناسایی کارکترها در هر ناحیه (یان^۱، ۲۰۰۸). برای کاهش میزان موفقیت تشخیص کارکترها توسط OCR سیستم کپچا از اعوجاج و درهم‌ریختگی در تصاویر استفاده می‌کند که از مراحل OCR جلوگیری کند. زمانی که حملات OCR قوی‌تری شود باید به کپچای متن نیز نویز و پس‌زمینه رنگی و خطوط اضافی و اشکال دیگر جهت جلوگیری از تشخیص افزود (هسو و لی^۲، ۲۰۱۱).

آزمون تورینگ برای اولین بار توسط التا ویستا در سال ۱۹۹۷ برای کاهش هرزنامه‌ها استفاده شد که کلمه‌ای به کاربر نشان داده و از کاربر خواسته می‌شد که آن را وارد کند.^{۱۵} در سال ۲۰۰۰ لوپس ون در دانشگاه ملون ایده کپچا را پیشنهاد داد و اولین بار در سایت یاهو برای جلوگیری از نفوذ ربات‌ها به چت روم‌ها به کار گرفته شد. کپچای gimpy توسط لوپس ون و بلوم پیشنهاد داده شد که به دلیل اینکه از لغات دیکشنری استفاده شده بود توسط موری و مالیک با استفاده از الگوریتم تشخیص تا ۹۹٪ به‌درستی تشخیص داده شد (بندی و شاه^۳، ۲۰۰۸). ریکپچا که توسط ون نیز پیشنهاد شده بود به دلیل استفاده از کلمات ناخوانا توسط OCR در نسخه‌های خطی دارای امنیت بیشتری بود (احمد و یان^۴، ۲۰۱۲). تکنیک‌های متن برجسته و دست خطی نیز ارائه شدند که تا حدودی کاربران را دچار مشکل می‌ساختند.^{۱۸}

۳. روش پیشنهادی مقاوم‌سازی با نقاط تلاقی

تعدادی پروژه تحقیقاتی تلاش کرده اند تا کپچاهای متنی را بوسیله‌ی برنامه‌های ی که دارای کارکردهای زیر هستند حل کنند:

پیش پردازش: حذف‌درهم ریختگی و پارازیت زمینه.

قطعه بندی: تقسیم تصویر به ناحیه‌هایی که هر کدام محتوی یک کاراکتر هستند.

طبقه بندی: تشخیص کاراکتری که در هر ناحیه است.

قدم‌های ۱ و ۳ برای رایانه‌ها بسیار ساده هستند. تنها قدمی که در آن انسان‌ها هنوز از رایانه‌ها قوی‌تر هستند قطعه‌بندی است. اگر درهم‌ریختگی زمینه از اشکالی شبیه اشکال حروف تشکیل شده باشد، حروف به‌وسیله‌ی این درهم‌ریختگی متصل شده باشند، قطعه‌بندی توسط نرم‌افزارهای فعلی تقریباً غیرممکن می‌شود. از این‌رو، یک کپچای مؤثر باید بر روی قطعه‌بندی تمرکز کند. ۳ پس در نتیجه این کپچا که بعد از عمل تقسیم‌بندی به‌راحتی توسط الگوریتم‌های OCR خوانده می‌شود و به‌راحتی شکسته می‌شود نیازمند تغییراتی ویژه است. تغییراتی که از ابتدا جلوی تقسیم‌بندی را بگیرد ولی باید توجه داشت که این عمل نیازمند ادغام و اعوجاج زیاد حروف است لذا باعث می‌شود که کاربر نتواند به‌سادگی آن کلمه را تشخیص دهد و دچار مشکل شود به همین دلیل ما در اینجا فقط بر روی جلوگیری از عمل قطعه‌بندی کپچا بحث می‌کنیم به‌طوری‌که با الگوریتم‌های ارائه‌شده قابل تقسیم نباشد

ادغام حروف

¹ Yan

² Hsu and Lee

³ Banday and Shah

⁴ Ahmad and Yan

ادغام حروف به معنی ترکیب و چسباندن حروف به هم تا جایی که به کلیت حروف صدمه‌ای وارد نشود و هرچند کلی قابل تشخیص باشند به طوری که حروف حداقل دارای یک اتصال باشند و فاصله خالی میان آن‌ها وجود نداشته باشد. به نقاطی که حروف به هم متصل می‌شوند نقاط تلاقی می‌گویند. این بخش دارای چند مرحله است:

مرحله اول درهم سازی حروف

از یک فونت مناسب جهت ایجاد چند حرف استفاده کرده. (می‌توانیم از فونت های دست خطی و سرهم نیز استفاده کنیم) ترکیب حروف کاملاً تصادفی است و نباید کلمه‌ی تولیدشده معنی‌دار باشد همانند شکل ۱.

ESL

شکل (۱): حروف E و S و L با یک فونت خاص

اگر عرض هر حرف Letter Length و فاصله درهم‌رفتگی دو حرف Letters Combination باشد درهم‌رفتگی به صورت زیر محاسبه می‌شود:

$$\text{Letters Combination} \geq \text{Letter Length} / 4 \quad (۱)$$

قبل از اینکه بخواهیم عمل درهم‌رفتگی را بر روی حروف اعمال کنیم محدوده قسمتی که حروف قرار است ادغام شوند را مشخص می‌کنیم مانند شکل ۲ تا با بتوانیم جدول تلاقی نقاط را ایجاد کنیم. پس از تست اندازه های مختلف می‌توان این مقدار را برای درهم سازی حروف مناسب دانست زیرا به اصل حروف صدمه‌ای وارد نمی‌شود، هرچند می‌توان در برخی از حروف درهم‌رفتگی را بیشتر کرد.

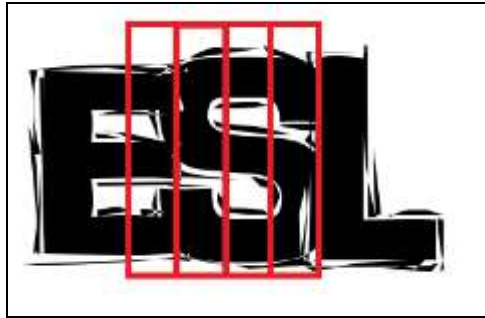


شکل (۲): تقسیم‌بندی حرف S

پس می‌توان نتیجه گرفت که فاصله‌ی بین حروف معادل عبارت زیر خواهد شد:

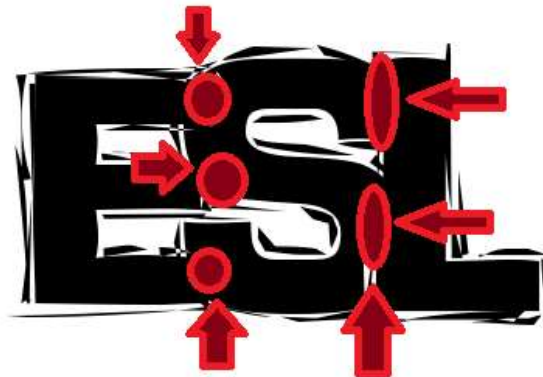
$$\text{Letter Space} = \text{Letter Space} - \text{Letters Combin} \quad (۲)$$

پس با توجه به رابطه بالا ادغام حروف انجام می‌شود بدین صورت که حروف با توجه به اندازه ادغام باهم ترکیب می‌شوند (شکل ۳) پس از ترکیب شدند بین هر دو حرف نقاطی به وجود می‌آید که نقاط تلاقی نام دارد (شکل ۴)



شکل (۳): قسمت‌بندی و ادغام دو حرف

باکم کردن فاصله بین حروف به اندازه حساب شده به صورت خودکار نقاط تلاقی شکل می‌گیرد و فقط باید این نکته را در نظر داشت که همه‌ی فونت‌ها دارای این خاصیت نیستند.



شکل (۴): تشخیص نقاط تلاقی در ترکیب ایجادشده

مرحله دوم ایجاد جدول تلاقی

حروف پس از عمل ادغام دارای نقاط تلاقی هستند که نحوه تشخیص آن‌ها در شکل ۴ مشخص شده. این نقاط تلاقی باید دارای خصوصیاتی هستند از جمله:

درهم‌رفتگی مشخص باشد و مرزی بین دو حرف نباشد

نقطه تلاقی در نقطه‌ای مشخص ایجادشده

خط مماس بین دو حرف به عنوان خط تلاقی معرفی می‌شود

باید مشخص و قابل تشخیص توسط چشم باشد.

هر حرف می‌تواند دارای یک جدول تلاقی دارای ۲۶ سطر باشد و در حالتی که حروف در حالت بزرگ را در نظر بگیریم یک جدول ۲۶*۲۶ از نقاط تلاقی برای حروف ایجاد می‌شود. خانه‌های این جدول به این صورت پر می‌شود که هر حرف با حروف دیگر از نظر نقاط تلاقی بررسی می‌شود و به تعداد نقاط برخورد در جدول عددگذاری می‌شود در صورتی که دو حرف بر روی یک خط باهم مماس شوند بدین معنی است که دو خط دارای خط تلاقی هستند و به جای عدد از علامت بی‌نهایت استفاده می‌شود. قسمتی از جدول در شکل ۷ نشان داده شده است.

Letter	A	B	C	D	E
A	1	1	1	1	1
B	1	∞	∞	∞	∞
C	1	2	2	2	2
D	1	2	2	2	3
E	3	3	3	3	3
F	3	2	2	2	2
G	2	∞	∞	∞	∞

شکل (۷): قسمتی از جدول تعداد نقاط تلاقی بین حروف

پس از ایجاد جدول تلاقی از آن استفاده کرده و قوانینی برای درهم سازی حروف وضع می‌کنیم. طبق جدول تلاقی حروف از حروفی استفاده می‌کنیم که دارای بیش از ۰ نقطه تلاقی و کمتر از ۴ نقطه تلاقی باشند و از حروفی که دارای خطوط تلاقی یا بینهایت نقطه تلاقی می‌باشند پرهیز می‌کنیم.

$$\text{Letter Join Point} \begin{cases} LJP > 0 \\ LJP < 4 \end{cases}$$

زمانی که حروف ادغام‌شده دارای نقاط تلاقی تحت شرایط فرمول ۳ باشند، خواندن حروف برای کاربر را نیز بسیار راحت‌تر می‌سازد. ادغام حروف به نحوی انجام می‌گیرد که دارای چند نقطه تلاقی باشند و دو حرف به‌طور کامل بر روی هم نیفتد. برخی از حروف دارای خط تلاقی هستند که اگر این حروف در کیچا در کنار هم استفاده شوند باعث ناخوانایی کلمه می‌شود (شکل ۸) پس باید در هنگام ساخت کیچا از در کنار هم قرار گرفتن این حروف جلوگیری کرد. عکس این عمل یعنی به کار بردن این حروف در کنار هم در کیچای پیشنهادی آینده می‌تواند مفید واقع شود و ضریب امنیت کیچا را بالا ببرد زیرا در عمل تشخیص توسط انسان خلی ایجاد نمی‌کند.

مرحله سوم اعمال بهینه‌سازی در حروف

جدول بهینه‌سازی جدولی است که جهت بهبود در روان خوانی کیچا به دلیل اعوجاج حروف و اتصال آن‌ها توسط کاربران ایجاد شده است. این جدول متشکل از حروفی است که برای بهینه کردن کیچا باید آن‌ها را تغییر داد تا کاربران آن‌ها را با حروف متشابه اشتباه نگیرند و به اشتباه پاسخ ندهند. (شکل ۹)

Let 1	Let2	Join	Join With Point
I	E	IE	E
H	E	HE	HE
J	E	JE	E
H	L	HL	HL
N	H	NH	NH

شکل (۸): ادغام های دارای خطوط تلاقی و مشکل ساز

حروف	متشابه	پیشن نمایش	حروف	متشابه	پیشن نمایش
vv	w	W	do	ob olo	ϕ
cl	d	d	ln	h	h
vW	WV	W	ob	do	ϕ
WV	VW	W	op	qe	ϕ
nn	m	m	lo	b	b
cm	an	m	bp	lop	lp
bl	lol	h	ld	lol	h

شکل (۹): نمونه‌هایی از حروف متشابه (۲)

مرحله چهارم اعمال پس‌زمینه و افکت

بعد از ساخت شدن کلمه کیچا با استفاده از مراحل بالا شروع به اعمال افکت گذاری و ایجاد اعوجاج در حروف می‌کنیم تا کلمه تولیدشده را از حالت یکنواختی و روتین بودن خارج سازیم باید توجه داشت که اعمال افکت نویز و اعوجاج کاملاً تصادفی است و از قانون خاصی پیروی نمی‌کند و فقط دامنه اعوجاج در حروف باید در بازه‌ای باشد که شکل حروف را دچار تغییر نکند و خوانایی حروف نیز حفظ شود. نویز پس‌زمینه نیز باید به‌گونه‌ای باشد که بتوان حروف را از پس‌زمینه تشخیص داد. (شکل ۱۰)



شکل (۱۰): کیچای متنی همراه با درهم‌رفتگی و افکت نویز و اعوجاج

شبه کد ایجاد کپچا

در ابتدا تعداد n حرف را که باید بین ۵ تا ۱۰ باشد به صورت تصادفی ایجاد می‌کنیم و در ارائه captcha ذخیره می‌کنیم حال حروف را با استفاده از جدول بهینه‌سازی بهینه کرده و سپس آن‌ها با فونت مورد نظر به صورت تصویر درآورده و نمایش می‌دهد. کارتهای ورودی توسط کاربر را با ارائه مقایسه کرده و درستی یا نادرستی آن‌ها تشخیص می‌دهد:

```

//////////////////////////////////Create Captcha:
n = Random Int From 5 to 10;
Array captcha [n]; // Number Of char In
Chaptcha is n
For (i= 1 to n)
    Captcha [i] = Random char ;
Optim(captcha); // Optimiza Arry With
Table
Graphic (g); // Definition Graphic to change
text to pic
For (i=1 to n)
    g.drawstring(captcha[i] , Newfont (AR
Darling)
return (g)
//////////////////////////////////Validate (input)
If (captcha.Equals(input))
    Return (True)
Else
    Return (False)

```

پیاده سازی

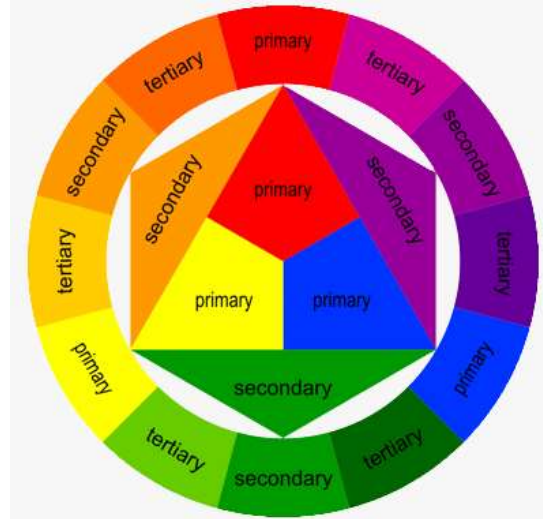
با استفاده از روند کار به صورت گفته شده به صورت مرحله به مرحله آن‌ها پیاده‌سازی می‌کنیم که خروجی کار به این صورت خواهد بود.



شکل (۱۱): پیاده‌سازی کپچای متنی

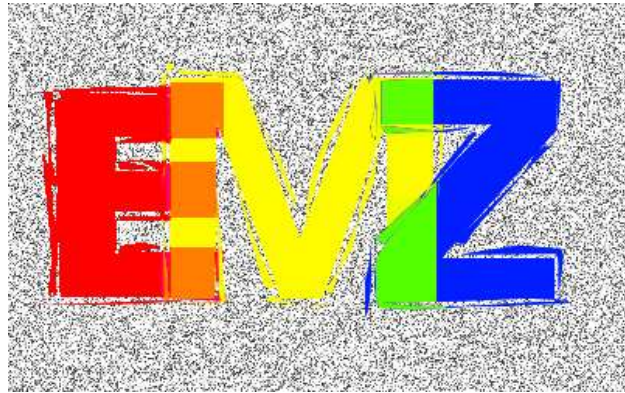
استفاده از ترکیب رنگ‌ها در کپچا

در ابتدا به تعریف رنگ‌های اصلی و فرعی می‌پردازیم تا بتوان درک بهتری از این روش پیدا کرد. رنگ‌های اصلی: در تئوری رنگ سنتی، قرمز، زرد و آبی رنگ‌های اصلی هستند زیرا دانه‌های رنگی تشکیل‌دهنده آن‌ها از ترکیب هیچ رنگ دیگری به وجود نیامده و تمام رنگ‌های دیگر از ترکیبات مختلف این سه رنگ با یکدیگر به دست می‌آیند. رنگ‌های فرعی: سبز، نارنجی و بنفش رنگ‌هایی هستند که از ترکیب رنگ‌های اصلی با یکدیگر حاصل می‌شوند. محل قرارگیری هر رنگ ثانویه در دایره رنگ، بین دورنگ اصلی تشکیل‌دهنده‌اش است. رنگ‌های ترکیبی: زرد-نارنجی، قرمز-نارنجی، قرمز-بنفش، آبی-بنفش، آبی-سبز و زرد-سبز، این رنگ‌ها از ترکیب یک رنگ اصلی با یک رنگ فرعی به وجود می‌آیند و در میان دورنگ تشکیل‌دهنده خود قرار می‌گیرند. (شکل ۱۱) حال با استفاده از مفاهیم ترکیب رنگ‌ها به طراحی کپچای قبل همراه با ترکیب رنگ می‌پردازیم.



شکل (۱۲): دایره رنگ‌های اصلی و فرعی و ترکیبی

کپچا را همانند بخش قبل طراحی می‌کنیم با این تفاوت که هر حرف دارای رنگی متفاوت در بازه رنگ‌ها خواهد بود و قسمت‌هایی که حروف باهم تلاقی دارند به رنگ ترکیبی از دورنگ خواهد بود (با توجه به شکل ۱۱) با استفاده از این روش می‌توان تشخیص حروف برای کاربر را بسیار افزایش داد. (شکل ۱۲)



شکل (۱۳): اعمال ترکیب‌بندی رنگ‌ها در کپچا

نتیجه‌گیری

در کارهای پیشین در بهبود کپچا با استفاده از رمزها و ترکیب رنگ‌ها یا حتی سه‌بعدی کردن کپچا سعی در جلوگیری از شکست و بهبود کپچا شده بود اما بعد از مدتی هرکدام با استفاده از روش‌های شکست تا درصد بالای ۹۴٪ شکسته می‌شدند. تنها کپچای مقاوم که خیلی خوب از سد شکستن عبور کرد، ریکپچا بود که اخیراً روش‌های جدیدی جهت شکستن آن تا درصد بالا ارائه شده است. ۷ جدیداً نرم‌افزار کپچا اسنایپر با استفاده از متد های شکست کپچا اکثر کپچاها را تا درصد بالایی شکسته و به راحتی از آن‌ها عبور می‌کند. گفتیم روش پیشنهادی با استفاده از نقاط تلاقی و کاهش خطای کاربر با استفاده از جدول تلاقی می‌تواند خود به‌تنهایی به‌عنوان یک کپچای مقاوم مورد استفاده قرار گیرد به طوری که همانند کپچاهای متنی موجود استفاده شود. در کپچای متنی پیشنهادی به دلیل وجود ادغام در حروف درصد شکست با الگوریتم‌های موجود به صفر می‌رسد ولی در نقطه مقابل کمی دقت بالای کاربر را می‌طلبد تا به درستی حل شود، اما همان‌طور که گفته شده دلیل ایجاد کپچای متنی مقاوم بالا بردن سطح امنیت کپچای تلفیقی است. در جدول زیر مقایسه چند کپچای موجود با کپچای متنی مقاوم از نظر درصد شکست در نرم‌افزار **Captcha Sniper** مورد آزمایش قرار گرفته که می‌توان این نتیجه را گرفت که هر چه درهم‌رتگی حروف در کپچا بیشتر باشد درصد تشخیص ربات پایین می‌آید به صوری که نزدیک به صفر می‌رسد ولی در نقطه مقابل کمی هم کاربر را دچار سردرگمی خواهد کرد که با استفاده از روش کپچای تلفیقی می‌توانیم از بروز این مشکل نیز جلوگیری کنیم.

Captcha	Percent detection	site
	97%	Pligg
	92%	phpBB
	87%	Joomla
	77%	Wordpress
	15%	nfe.fazenda.gov.br
	6%	TubeX
	1%	Cocaptcha

شکل (۱۴): مقایسه درصد شکست کپچای پیشنهادی با سایر کپچاها در نرم‌افزار **Captcha Sniper**

در روش پیشنهادی از انواع فونت می‌توان استفاده کرد و می‌توان با استفاده از علائم ریاضی و پرکاربرد درون کپچا به بالا بردن امنیت آن نیز کمک کرد تا درصد شکست آن را به صفر برساند. باینکه هر چه درهم‌رفتگی حروف بیشتر شود امنیت بالا می‌رود و غیرقابل شکست می‌شود ولی در نقطه مقابل نیز برای کاربر غیرقابل تشخیص می‌شود پس باید حد بین را رعایت کرد که از حالت تعادل نیز خارج نشود. به‌عنوان پیشنهاد می‌توان گفت که با استفاده از ترکیب رنگ‌ها در حروف حتی می‌توان تا میزان بیشتری حروف را در هم برد به‌گونه‌ای که برای کاربر مشکلی ایجاد نکند.

مراجع

1. Ahn, L. Von. Blum M., Hopper N. (2004). CAPTCHA: Telling humans and computers apart automatically, communication of the ACM, Vol. 47, No.2
2. Alsuhibany, S. A. (2011, 22-26 Aug. 2011). *Optimising CAPTCHA Generation*. Paper presented at 2013 International Conference on the Availability, Reliability and Security (ARES).
3. Chandavale, A. A. & Sapkal, A. M. (2010, 19-21 Nov. 2010). *Algorithm for Secured Online Authentication Using CAPTCHA*. Paper presented at 2010 3rd International Conference on the Emerging Trends in Engineering and Technology (ICETET)
4. Kaur, K., & Behal, S. (2015). Designing a Secure Text-based CAPTCHA. *Procedia Computer Science*, 57, 122-125. doi:http://dx.doi.org/10.1016/j.procs.2015.07.381
5. Ling-Zi, X. & Yi-Chun, Z. (2012, 10-12 Oct. 2012). *A Case Study of Text-Based CAPTCHA Attacks*. Paper presented at 2012 International Conference on the Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC).
6. Obimbo, C., Halligan, A., & De Freitas, P. (2013). CaptchAll: An Improvement on the Modern Text-based CAPTCHA. *Procedia Computer Science*, 20, 496-501. doi:http://dx.doi.org/10.1016/j.procs.2013.09.309
7. Starostenko, O., Cruz-Perez, C., Uceda-Ponga, F., & Alarcon-Aquino, V. (2015). Breaking text-based CAPTCHAs with variable word and character orientation. *Pattern Recognition*, 48(4), 1101-1112. doi:http://dx.doi.org/10.1016/j.patcog.2014.09.006
8. Wexler, M. (2001). Robots help humans defeat robots. *TRENDS in Cognitive Sciences*, 5(12), 512. doi:http://dx.doi.org/10.1016/S1364-6613(00)01824-6
9. Prof. Yogdhar Pandey and Darshika Lothe, "Evaluating the Usability and Security of a Spelling Based Captcha System," *International Journal of Computer Science and Information Technologies*, Vol. 5 (3), 2014, 4728-4731.
10. M. Chew and J. D. Tygar, "Image Recognition CAPTCHAs", In *Proceedings of the 7th*
11. Von Ahn Luis, B. Manuel and L. John, "Telling Humans and Computer Apart Automatically," *CACM*, V47, No2.
12. Clark Pope and Khushpreet Kaur, "Is It Human or Computer? Defending E-Commerce with Captcha," *IEEE Computer Society*, March –April, IT PRO 2005.
13. J. Yan, A.S.E. Ahmad, "Usability of CAPTCHAs or usability issues in CAPTCHA design," Presented at the 4th Symposium on Usable Privacy and Security, Pittsburgh, Pennsylvania, 2008.
14. Chih-Hsiang Hsu and Ying-Lien Lee, "Usability study of text-based CAPTCHAs," 2011Elsevier, vol.32 (2011) 81–86.

15. J.Ogijenko and A.Kolupaev, "Captchas: Humans vs Bots." IEEE Computer Security, vol.6, pp.68-70, Feb. 2008.
16. M. Tariq Banday and Nisar A. Shah, "Drag and Drop Image CAPTCHA." Sprouts 4th J&K Science congress, 2008 .
17. Ahmad El Ahmad, Jeff Yan and Wai-Yin Ng , "CAPTCHA Design Color, , and Security," IEEE Computer Society, 1089-7801, March – April 2012 IEEE.
18. N. A. Shah and M. Tariq Banday, "A Study of CAPTCHAs for Securing Web Services." IJSDIA International Journal of Secure Digital Information Age, Vol. 1. No. 2, December 2009.