Generation of Authentication Strings From Graphic Keys

Sérgio Tenreiro de Magalhães[1], Kenneth Revett[2], Henrique M. D. Santos[1]

[1] University of Minho
Department of Information Systems
Campus de Azurem
4800-058 Guimaraes, Portugal
{psmagalhaes, hsantos} @dsi.uminho.pt

[2] University of Westminster
Harrow School of Computer Science
London, UK HA1 3TP
revettk@westminster.ac.uk

Abstract:
The traditional authentication system used in technological applications is the well-known and widely spread user/password pair. This technology as proved itself as well acceptable by the users and quite safe when used according to good security practices, this is: frequent change of the password; use of letters, number and symbols in the password; not revealing the password to others; not using the same password in more then one service; etc. But this is not what really happens, so we need to improve the protocol. Graphical secrets present lots of advantages and can increase the level of security without a significant change in the users habits. For that, we need to possess strong ways to convert them into strings that will fed the implemented passwords systems. In this paper we present a method to do so.

Keywords: Authentication; Graphic Keys, Passgraphs; Key Space; Password creation.

Introduction

In the context of Information Systems (IS), authentication is the process of confirming an alleged identity and it differs from the identification process in which a user is linked to a known identity [1]. Authentication involves, traditionally, sharing a secret with the authenticating entity and presenting it whenever a confirmation of the user's identity is needed. In the digital era that secret is commonly a username/password pair and/or, sometimes, a biometric feature, both presenting difficulties of different kinds once the first has known vulnerabilities and the second has many issues related to ethical and social implications of its use [2].

Password vulnerabilities come from their misuse that, in turn, results from the fact that they need to be both easy to remember, therefore simple, and secure, therefore complex. Consequently, it is virtually impossible to come up with a "good" password [3]. On the other hand, once users have not yet completely realized the need for securing their authentication secrets, even fairly good passwords become a threat when the security policies (if at all existing) fail to be implemented. The results of an inquiry made by the authors in 2004 to sixty Information Technology (IT) professionals show that, even among those that have technical knowledge, the need for passwords security is underestimated.

Table 1 - The distribution of the passwords constitution shows a generalized vulnerability

Constitution of the passwords	Users (%)
Letters and symbols	0%
Numbers and symbols	0%
Letters, numbers and symbols	17%

Only letters	23%
Only numbers	17%
Letters and numbers	43%

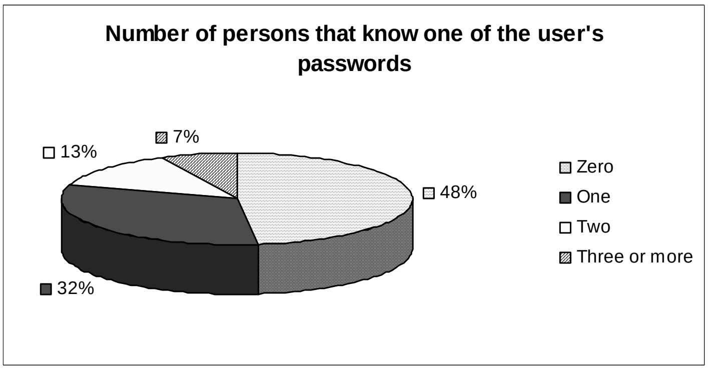


Figure 1 - Users have a generalized tendency to share their passwords

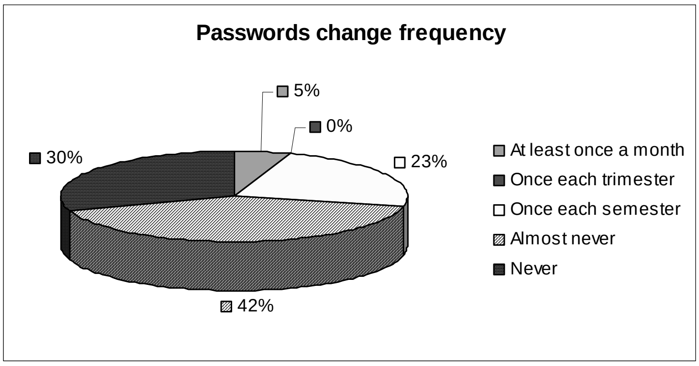


Figure 2 - Most of the users rarely change their passwords

As shown in the table 1, only 17% of the inquired professionals use complex codes including symbols, and 72% stated that they rarely change their access codes (figure 1), despite 52% of them know that at least one of those is known by at least one other person (figure 2). This need for simplicity and the principle of trust that allows a user to have the password on a post-it placed under the keyboard or even on the monitor, creates a security breach that can be stopped by graphical secrets (passgraphs), once they are easier to remember [9] [10], they can generate complex passwords (an easy way to assure easy compatibility with existing systems) and they are difficult to transmit from person to person. This need to stop the transmissibility of the authentication secrets is even bigger when we realize (figure 3) that most professional users (65%) have only one or two codes that they use for authenticating to the generality of the used services.

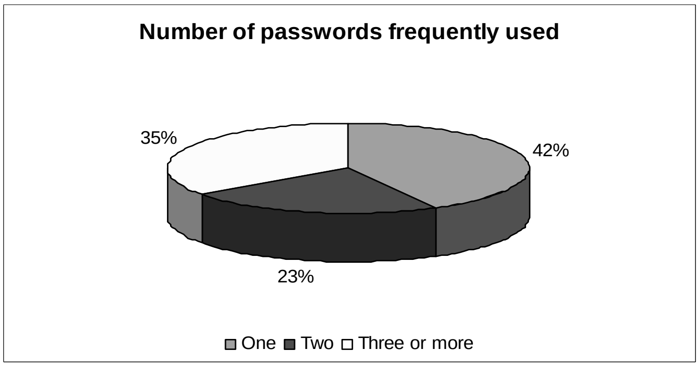


Figure 3 - Most of the user use the same password for accessing all services

Needless to say that the authentication processes based on passgraphs are, like virtual keyboards, adequate for use in private spaces or in small devices like the Personal digital Assistants (PDAs), once they are vulnerable to eyesdroping. Giving to the user the possibility to choose at each login attempt between the passgraph mode and the password mode is not a choice either, once the system would inherit the vulnerabilities of both systems, so the only way to implement this systems without limiting the users is to allow the user a choice between the system when he uses the system for the first time (enrolment) and making that choice definitive (or almost). In this case, the user must be educated for the advantages and disadvantages of both systems so that he can make the choice that best suites is needs. Anyway, in order to provide an easy widespread of passqraph systems and to take advantage of the security infrastructures already deployed they must be compatible with the existing systems, without generating new vulnerabilities. This can be achieved by generating strong passwords from the users passhraph choices using the one-way function described on this paper.

Previous work on passgraphs

Greg Blonder was the first to describe graphical passwords [4], presenting in a United States Patent a system that would allow users to choose their picture, the number of regions to be clicked, their size and position. Since then, many variations of this system were presented and images have gained their way into the authentication processes.

Among the most popular graphical authentication systems we find Passfaces™ from the Passfaces Corporation, a commercial system were the user chooses a previously selected face from a set of faces and repeats this process for different faces in different sets for a defined number of times [5], but popular doesn't imply secure and a study of the users choices demonstrated that they are, in some cases, similar for all users. For instance, 10% of the passwords of males could have been guessed with only two attempts [6].

The *Déjà Vu* Scheme inn which a matrix of *m* images is set, where *n* images are part of the user's portfolio, previously chosen from a set of proposed images. The user must identify those *n* images to login.

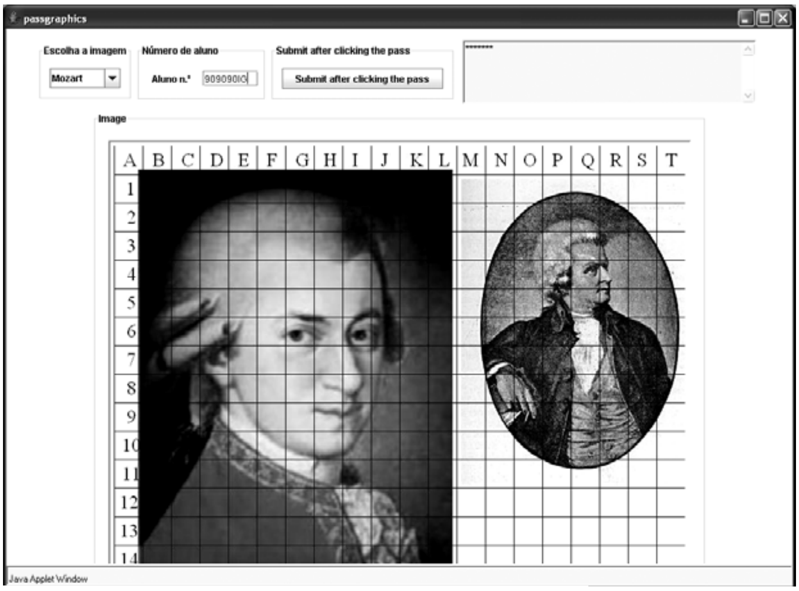
The *draw a secret* (DAS) scheme is a graphical authentication system with an approach completely different. In DAS the user draws something over a grid and that becomes is authentication secret. This system has been implemented with success in PDAs and further studies will be made to analyse the user's choices and acceptance [7].

In the Visual Identification Protocol (VIP) several possibilities were created. From a set of ten predefined images the user chooses four, placed on the same place and typed in the same order (VIP1) or placed in random positions (VIP2). VIP3 is a process where four of the eight images existing in the user's portfolio are displayed along with 12 distractors and the user must identify them in no particular order. The studies shown that the most common errors associated with VIP1 and VIP2 were related with bad sequences, when the identified images are correct but selected on the wrong order, and in VIP3 most of the errors were due to wrong identification of the images, for instance any flower being consider as “the” chosen flower [8].

Description of the implemented system

Considering that the PDA is the technology that provides an environment that better takes advantage of the graphical authentication procedures and that the Word Wide Web is the most used distributed system, our system was designed to meet the authentication demands of a Mobile Web Service, especially web pages destined to be browsed in PDAs.

In order to test the authentication process in a real life situation we transformed the login procedure of the site of a graduation course, normally protected by password for copyright reasons and to protect the privacy of the students in matters like their grades. We use the Web Application Server PE 8, from SUN, for a regular file realm authentication invoked through a hidden field in a regular form. The created environment feeds this field with a string that results from the application of a function (described later) to the passgraph data. The security issues were then addressed in the same way that for a regular username/password authenticated site.



Escolha a imagem: Mozart

Número de aluno — Aluno n.º 90909010

****

passgraphics
Submit after clicking the pass
Submit after clicking the pass
Image
A B C D E F G H I J K L M N O P Q R S T
1 2 3 4 5 6 7 8 9 10 11 12 13 14
Java Applet Window

Figure 4 - The enrolment and authentication window

Both the enrolment and the authentication environment consist on a window with a field for a username, in this case the student number; a dialog panel, where the users get feedback from the system; and a select panel, where the user can choose the active image from a previously defined set. Each image includes a grid in which each area can be seen as a pair (letter, number) once the first line has letters and the firs column has numbers letter A in the first position, equivalent to the line zero. In the first stage the image was divided in a 20x15 grid (figure 4) but, now that our community is getting familiar with passgraphs, we'll use a cc grid in the next implementation. This way we can place the entire alphabet on the first line. This configuration was chosen in order to find out if the users would prefer a password scheme, instead of a graphical scheme. If this were the case, the user would just pick a sequence of letters. On the other hand, this can also be used (with a full alphabet) for entering the username without the use of an external keyboard (virtual or physical). The user chooses an authentication secret by clicking on several points of the active image, which can be changed at any time. The sequence of the picked points in the chosen images will be the authentication key for future logins. The first implementation was conceived to fully understand the users free choices; therefore no limitations were imposed to the length of the passgraph. In the dialog panel the symbol * appears whenever the user selects a region of the image, allowing it to know if he did in fact clicked or if he accidentally clicked twice. In this way we'll be able to reduce the number of login errors due to accidental clicks. In the final enrolment implementation, this panel will also transmit information about the limitations that will be made to the length of the passgraph. The new web site has now a limitation to the size of the passgraph, being four the smallest selected sequence.

The process

The use of several images (so far we have always used the same four images) creates a third dimension factor once a passgraph with length *n* will be a vector of the type (p_1, p_2,…, p_n) where, considering I as the set of available images, $p \in \{(x, y, z) \mid 0 \le x \le 19,\ 0 \le y \le 13,\ 1 \le z \le \#\mathrm{I}\}$. The values of *x* and *y* define the selected section in a specific two-dimensional image, as shown in figure 5 for a set of four images (the ones used in our implementation). In order to maintain the compatibility with the traditional password systems, we need to generate a string. For that, we'll use 15 tables, numbered from 0 to 14, with 26 columns and 20 lines. So we have 7800 cells, each one with one 3 characters string and the corresponding ANSI code. Figure 5 is one of those tables.

0																									
4kQ	v9;	D't	~\'	M'p	K"|	.n.	@8v	(!p	LQi	H=p	j,6	h$m	,:]	q;_	]h8	D-{	-Yh	`Rj	T^-	evf	y6:	i*f	jJH	^sl	{{T
E6W	xnf	:}V	Sx8	e(r	XT0	;8u	aWA	x6w	pd/	#,T	t4j	EwB	;!^	ElX	VOU	PjE	br4	wch	;!I	A:/	"eb	D,S	8R1	8%j	dQi
B`o	}5N	yDv	y$U	;3t	F`|	[+-	X!g	yvd	HdF	9XQ	Wcp	2-u	pEB	}Tj	jZI	_mx	f#I	5Df	&)^	?d$	up6	3:@	u=_	r<_	yjK
T~3	LZM	hF7	.'V	]`d	C}{	-5%	];3	qwn	&!n	-vf	sTr	IrV	X3d	aZn	zs4	HqV	l1U	@v>	AUs	L[a	?/n	OkA	|S/	=s/	j<6
ZU8	xtx	yT2	ekN	Evy	V^#	)I2	2Fv	<w&	u{=	I6L	?f0	\r.	Jew	6I/	zL]	`Oc	zJk	Ati	Y&h	xG\	O]^	L3E	P#D	8.d	tfx
!Z[	He$	'Ot	HdS	QB(	>,y	ewC	}{d	ayq	%+"	a"}	0LY	_DE	NjD	53I	xXV	ry6	~,Y	;l6	({e	$,8	D~2	,Mp	IL9	-zI	Ad~
aC&	@,H	jos	Hg4	e$Q	(i?	0f|	k;n	hpS	Dx$	z0&	?P3	W#f	k:B	&k`	v#^	O/t	0)q	48a	4~!	sv\	N.!	gZb	~"J	?{<	zaL
_~t	0lp	:%S	%$K	bUb	`f-	`9(	3"'	u<W	HKI	@m=	CTQ	O)H	no;	A-t	{l#	Q\)	rGc	1TX	|Vy	`<o	I7B	~Np	","	3^$	gYr
P?p	x7U	&^"	FLn	F|.	,\b	1jo	4{E	L$4	_##	6Bv	h8]	e@J	KFP	cO&	}SN	$kS	fN!	b(!	-%w	i]/	c7`	#P:	;an	2=!	q^4
k2"	v0n	z$S	v4j	W?,	,yL	$#u	-q)	&O}	wZt	\G|	]5B	k\a	BK6	Xhp	m=c	[(Q	Tkd	+!r	!Y^	IMf	KVf	jjM	rjJ	tF#	+v(
-md	eYY	t4Y	ali	vI!	cT6	MI@	wsk	8Bo	jES	|9R	#&.	N.k	;??	(*V	h*i	0@^	^1"	xB$	-lv	<.t	Kb/	V9<	&2m	;.0	D#r
ZVW	j.W	X~O	/f3	g@5	p%h	rAM	kuv	B[9	xzb	UQR	4"b	gIN	S6*	@6}	j]6	UMr	z*H	n.K	O-H	8el	<>r	4H2	t[/	p;P	ZVw
WKA	.y>	?"B	XDw	+q,	OQ"	DNL	]d1	9If	A[7	/Y'	j8\	h|1	CF*	DV:	tot	(>~	}4H	y%3	V[o	_*s	!5Q	ey&	$o\	d(%	o{@
]04	e5Y	$rB	&q@	~~E	?<!	3[/	ZG_	"MD	9t[	{DF	|zU	D`d	6B~	Z?|	tjI	8o+	RFi	~W3	s7F	?Sj	]V[	`RS	.5q	xke	dgQ
"M:	OCn	G%k	wyv	R~o	9kb	7uM	gCa	,|,	2G1	'Ew	CFi	Ljz	v+d	#1&	/e$	oxG	-o2	U`j	3k!	<2q	#wE	<ub	Z2T	Ig$	zK'
ggz	t4%	\a:	J\2	{[_	ky>	|kI	K07	&6x	:N:	>5V	}eg	>L2	2su	Cb^	M)(	3?T	Z|*	-^D	gUr	>LI	=C:	rMS	b^~	>6/	pte
W&H	!bQ	:Ww	dbX	P6U	AkE	`7K	G#%	u6V	TN9	A:3	1Q(	:L5	Gm-	@HA	'jc	Zs[	>N/	c;?	n,>	0rW	5}}	qQ(	B3I	*Bh	#Ti
!~z	UxU	veV	H~?	;b#	S80	:=?	HXd	^H~	y!1	Tt*	.db	V}7	Ff!	XYt	P=W	iKO	t-3	w1|	ubN	Wsk	\Ko	W8x	vbu	+q2	]),
Ud.	@yL	cc[	tba	1wP	sXB	LGk	-'A	sSC	dxU	~6Z	A6r	7[m	1v%	R'M	"7]	~`k	W&5	IXI	62]	tDn	V#R	LLj	pHp	XHm	v=G
}C!	a:E	<qM	QG#	w(@	M&,	[<@	!QP	yn*	*D~	s2W	y4@	VJ'	do*	TU"	,<B	SRA	x9t	K,I	DzC	_6-	S,8	~Q]	).?	N^}	hVY

Figure 5 – One of the conversion tables.

We find our first cell by locating the table z in line x and column y. Then, for each $p \in \{(x, y, z) \mid 0 \le x \le 19,\ 0 \le y \le 13,\ 1 \le z \le \#\mathrm{I}\}$ we'll do:

- $(x + y + z + 1^{st}\ ANSIFromThe\,\mathrm{Pr}\,eviousSelectedCell) \bmod NumberOfLines$ to find the next line selected;
- $(x + y + z + 2^{st}\ ANSIFromThe\,\mathrm{Pr}\,eviousSelectedCell) \bmod NumberOfColums$ to find the next column selected;
- $(x + y + z + 3^{st}\ ANSIFromThe\,\mathrm{Pr}\,eviousSelectedCell) \bmod NumberOfTables$ to find the next table selected;

To prevent the possibility of discovering the sequence of clicks from the string if this is compromised, for instance by capturing the packages on a non-encrypted network, we need to make some final changes in the string. In this way, frequent changes in the tables (and the correspondent passwords) can increase the level of security of the system, in a transparent way to the user that will continue to click in the same places of the same figures.

Let x be the ANSI code of the first element of the so far generated string. Given *t* = x *mod* n, will reverse the order of the first *t* characters.

Let y be the ANSI code of the last element of the so far generated string.. Given k = x *mod* n, will reverse the order of the last k characters.

For instance the generated string from a simple sequence of clicks, shown in Figure 6, is ***p"$p}sLQyNUi***.

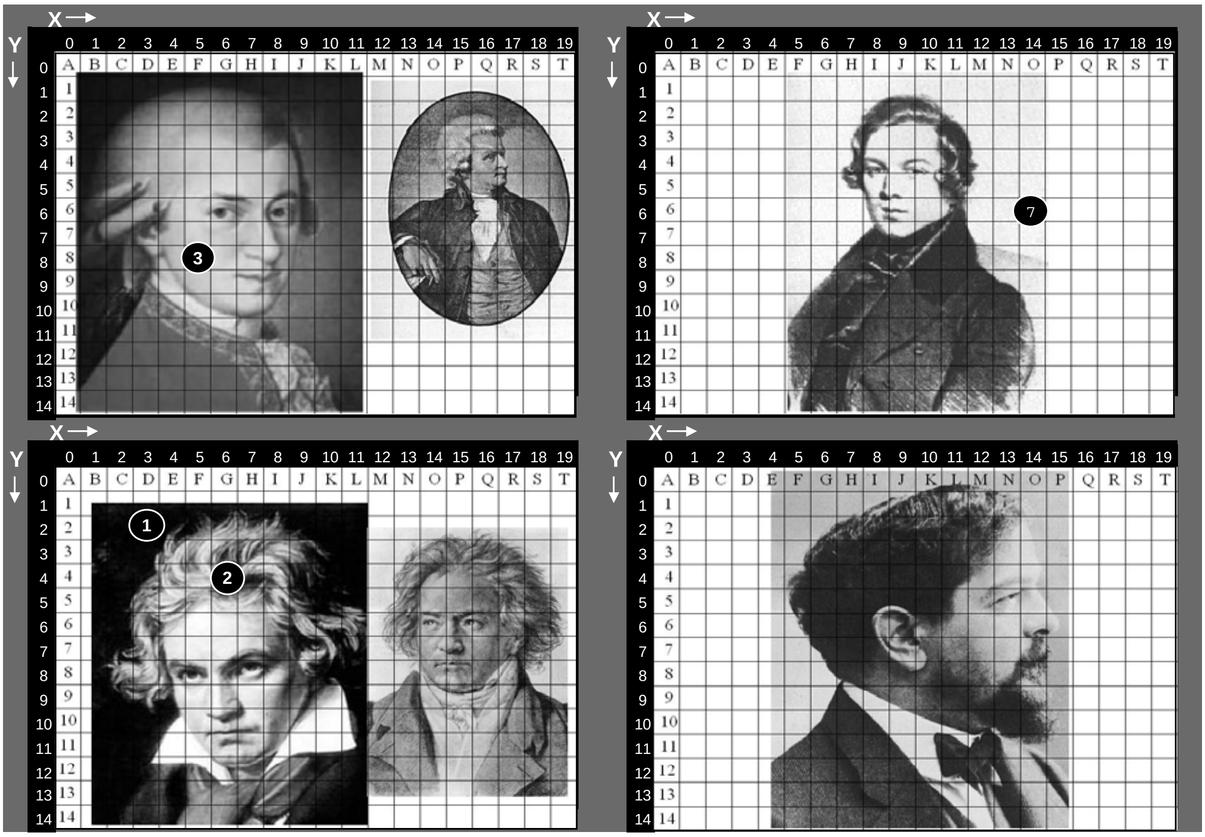


Figure 6 – The shown passgraph will generate the password *p"$p}sLQyNUi*.

Conclusions

In this paper we present a method to generate strong authentication strings, usable in common user/password authentication systems. The process can generate different passwords with simple transformations in the conversion system and in the password files, in a transparent way to the user that will continue to use the usual graphical secret.

Once again, we verify that the graphical secrets can be used in authentication with strong advantages to what concerns to security and without significant entropy to the users.

Bibliography

[1] Magalhães, S. T., Revett, K. and Santos, H. D.: *Password Secured Sites - Stepping Forward With Keystroke Dynamics*, Proceedings of the IEEE International Conference on Next Generation Web Services Practices, IEEE CS Press, Seoul, South Korea, 2005.

[2] Magalhães, S. T. and Santos, H. D.: *An Improved Statistical Keystroke Dynamics Algorithm*, Proceedings of the IADIS Virtual Multi Conference on Computer Science and Information Systems, 2005.

[3] Wiedenbeck, S., Waters, J., Birget, J.C., Brodskiy, A. and Memon, N.: *Authentication using graphical passwords: Basic results,* Human-Computer Interaction International (HCII 2005), Las Vegas, July 25-27, 2005

[4] Blonder, G. E.: *Graphical password,* U.S. Patent Number 5.559.961, 1996.

[5] The science behind Passfaces^{TM}

[6] Davies, D., Monrose, F. and Reiter, M. K.: *On User Choice in Graphical Password Schemes*, 13th USENIX Security Symposium, 2004.

[7] Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K. and Rubin, A.: *The Design and Analysis of Graphical Passwords*, ??, 1999

[8] De Angeli, A., Coventry, L, Johnson, G.I and Coutts, M.: *Usability and user authentication: Pictorial passwords vs. PIN,* In P.T.McCabe, (Ed.). Contemporary Ergonomics 2003 (pp. 253-258) London: Taylor & Francis, 2003.

[9] Nelson, D. L., Reed, U. S. and Walling, J. R.: *Picture superiority effect,* Journal of Experimental Psychology: Human Learning and Memory, 3:485–497, 1977.

[10] Madigan, S.: *Picture memory*. In Imagery, Memory, and Cognition, pages 65–86, Lawrence Erlbaum Associates, 1983.