

Program Title Unbreakable Cipher System; Cryptography

Contributor's Name John G Schmid (WA6PGA)

Address 1528 West Cherry Avenue

City Lompoc State CA Zip Code 93436

Program Description, Equations, Variables There is a very interesting résumé describing a truly unbreakable cipher system; the local library should have a copy of the 8/77 issue of Scientific American (p 120). This HP67 program also uses the shift cipher principle; it calls for deciding upon 1 "Initiator" number and 9 "Key" numbers, each having 10 places, which are then used in the encoding and decoding process. Each Key consists of a number between 1 and <10 and the Initiator lies between 1 and <100; the integer part of the Initiator is given by the number of characters in the chosen alphabet, up to & including 99. The 10 places of the 9 Keys and the eight decimal places of the Initiator are randomly chosen and agreed upon; each can be anywhere from 0 thru 9 and is independent from any other. The Keys and the Initiator fraction part are modified for each character process such that the cyclical shift principle becomes a pseudo-random shift process. In addition, to make "cracking" the code all the harder, if not practically impossible, the clear character numeral equivalent of the previous entry or decoding result is also used to further modify the randomizing process: In other words, when one has an unknown encrypted text, he is ignorant of the previous clear numeral; since he needs it, however, the breaking of the code is a good challenge to a member of the cryptanalyst breed! This also results in the quaint characteristic that a wrong guess immediately causes all the following numerals to go wild in an exceedingly complicated manner. Note also the action of Program Step 46 as well as 13, permitting the operator further deviations; there are several other possibilities to change the Program that result in rather hairy ciphers, or "improvements" on the present one. As an example, consider replacing Program Step 95 with the following:

RCL m, RCL n, +, RCL p, +, g FRAC, h y^x (which will bump up the remaining steps to fill all 224 steps)
 where m, n, p can be any integer from
 0 thru 9. Since the 10 primary registers change with every character processed, some more non-constant imponderables are thrown into the calculations.

Operating Limits and Warnings This program is of course not of the Diffie/Hellman/Rivest/Shamir/Adleman calibre. The machine can in no way handle such monstrous numbers as they are routinely playing with. However, with this many combinations, permutations, and wild guesses possible with the 9 Keys and the Initiator, plus the idea of the previous-clear-numeral modifier, I wonder if the following message will get unriddled:
 34942 42364 10629 32300 21429 47402 22519 17251 54404 04240 22830 30473 34747 14442
 Sorry, no help is offered other than the fact that an HP67 was used.
 03811 03454 34710 02141 60907 92445

This program has been verified only with respect to the numerical example given in *Program Description II*. User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP NOR THE CONTRIBUTOR MAKES ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND WITH REGARD TO THIS PROGRAM MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER HP NOR THE CONTRIBUTOR SHALL BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, USE OR PERFORMANCE OF THIS PROGRAM MATERIAL.

Program Description II

I Sketch(es) (1 A R A T 2 B K L : 3 G O R 4 C R D N P 5 E 6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
P	W	X	H	7	F	I	G	A	+	H	8	V	I	T	J	S	K	L	F	U	Q	9	M	L
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
50	51																							
2	N		0	7	Y	4	ø	Z	8	P	Q	5	R	Z	?	S	T	B	U	V	C	-	A	R
52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76
77	78																							
B	U	A	V	C	1	ø	D	Y	V	F	E	N	A	O	G	)	I	M	E	/				
79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99				

SAMPLE
ALPHABET

II Sample Problem(s) Encode following text using above ALPHABET and below KEYS and INITIATOR:

EXAMPLE OF COMPUTER - GENER

54 54 54 23 85 92 97 62 49 98 54 93 44 54 83 93 97 21 71 39 90 65 74 94 98 53 23 76

33 99 28 18 80 18 79 56 02 25 67 23 26 03 91 91 56 20 70 46 47 08 33 89 57 65 39 38

A T E D C Y P H E R : D O N E W I T H H P 6 7 /

75 06 90 86 54 17 57 62 28 23 15 11 54 86 55 91 23 54 78 31 06 35 54 28 25 24 56 99

01 67 32 43 57 50 08 98 18 43 37 78 64 30 08 67 57 42 59 25 66 35 01 70 81 24 73 73

1 7 0 2 A ø ø ø 3 3 .

02 29 59 52 75 59 57 59 12 12 50 54 54 54

48 14 18 48 79 58 93 86 36 86 34 37 29 60

III

Initiator (Key ø)	Key 1	Key 2	Key 3	Key 4	Key 5	Key 6	Key 7	Key 8	Key 9
99.74285369	3.852	964	712						
	2	5.471	892	365					
	3	7.829	531	645					
	4	6.142	798	653					
	5	4.517	965	329					
	6	2.718	469	325					
	7	1.728	953	645					
	8	9.100	578	425					
	9	7.250	840	935					

IV Reference(s) — N/A

(PS: If the repeating Pause of Step 134 makes you nervous, simply replace it with an R/S: However, it then becomes necessary to end every Step 10 or 14 of the Instructions with a manual R/S "continue" command.)



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Agree upon the "Alphabet" that will form the basis of the coding process. Go to 2.	[p.2, SKETCH]		
2	Agree upon the "Initiator" and the 9 "Keys". Go to 3.	[p.2, Block 3]		
3	Run in both sides of Program card. Go to 4.			Ø.
4	If Data card is preloaded with the appropriate data (Initiator and Keys), press no keys, go to 5 directly. If not, Key in: Initiator number	< 100 pos.	A	1.
	Key # 1	< 10 pos.	R/S	2.
	# 2	"	"	3.
	3	"	"	4.
	4	"	"	5.
	5	"	"	6.
	6	"	"	7.
	7	"	"	8.
	8	"	"	9.
	9	"	"	Crd
	Go to 5; or [press any key] and go to 6.	{ nil	[any]	[Ø]
5	Run in Data card side 1. Go to 6.	{ optional		Ø.
6	"Translate" the message/text into numerals in accordance with the Alphabet mentioned in Step 1. Go to 7.	[see p.2, lines a and b]		
7	For <u>encoding</u> , go to 8; <u>decoding</u> to 12.			
8	Take the first translated character of the text (whether it previously was letter, numeral, space, punctuation, whatever) and key that number in. Press E (for <u>Encode</u>). Go to 9.	[p.2, line b]		
		≤ A pos.	E	...
9	The new number showing during the repeating Pause is the encoded number equivalent of the text's original character. See Note 3! Write it down (to start/continue the string of code numbers that result from continuing encoding). Go to 10 if continuing; else go to Step 11.	[see p.2, line c]		... rep'g

- optional -

Instructions Step 5
Program Step 50

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
10	During one of the Pauses, key in the next translated character. Go to 9.	[p.2, line b] ≤ [A], pos.		
11	When coding is complete, press B to Blank out all revealing data contained in machine except program.	END.	B	φ.
12	Key in 1 st pair of numbers (such as 37, or 43 or 19 or etc.) and press D for Decode. Go to 13.	≤ [A], pos. [p.2, line c]	D	...
13	The new number showing during the repeating Pause is the "translation" number of the clear text's corresponding character. Note 3. With help of the Alphabet then translate it back to the clear character equivalent. Write it down to start/continue the string of clear text characters. Go to 14 or 11 as applicable.	[p.2, lines b → a]		
14	During one of the repeating Pauses, Key in the next code number pair. Go to 13.	[p.2, line c]		
Note 1:	The agreed-upon total number of characters in the Alphabet is anything up to incl. 99.			.
Note 2:	Changing just one digit of any Key affects the coding process to result in a totally different string of code numbers.			
Note 3:	To review the number last keyed in, press "figure check" fc: To verify "serial" number of the number now showing, press "Count": Continue only after clear or code number shows again.		f c C	

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	• FLBL A	31 25 11			EEX	43	
	h CF 2	35 61 02			3	03	
	Sto A	33 11			h π	35 73	
	g frac.	32 83		060	—	51	
	Sto B	33 12			g frac.	32 83	
	1	01			Sto C	33 13	
	$\emptyset$	00			rcl 8	34 08	
	X	71			f GSB 8	31 22 08	
	f int.	31 83			Sto + 8	33 61 08	
010	9	09			rcl 7	34 07	
	h $x-y$	35 52			f GSB 8	31 22 08	
	g $x=y$	32 51			Sto + 7	33 61 07	
	3	03	Can be any number from $\emptyset$ thru 8.		rcl 6	34 06	
	h $x-I$	35 24		070	f GSB 8	31 22 08	
	Rcl A	34 11			Sto + 6	33 61 06	
	f int	31 83			rcl 5	34 05	
	Sto A	33 11	number of Alph. Char.		f GSB 8	31 22 08	
	1	01			Sto + 5	33 61 05	
	R/S	84			rcl 4	34 04	
020	Sto 8	33 08	Key 1		f GSB 8	31 22 08	
	2	02			Sto + 4	33 61 04	
	R/S	84			rcl 3	34 03	
	Sto 7	33 07	Key 2		f GSB 8	31 22 08	
	3	03		080	Sto + 3	33 61 03	
	R/S	84			rcl 2	34 02	
	Sto 6	33 06	Key 3		f GSB 8	31 22 08	
	4	04			Sto + 2	33 61 02	
	R/S	84			rcl 1	34 01	
	Sto 5	33 05	Key 4		f GSB 8	31 22 08	
030	5	05			Sto + 1	33 61 01	
	R/S	84			rcl 0	34 00	
	Sto 4	33 04	Key 5		f GSB 8	31 22 08	
	6	06			Sto + 0	33 61 00	
	R/S	84		090	rcl 9	34 09	
	Sto 3	33 03	Key 6		h absol.	35 64	
	7	07			rcl C	34 13	
	R/S	84			X	71	
	Sto 2	33 02	Key 7		g frac.	32 83	
	8	08			f $\sqrt{x}$	31 54	can be any fractional power, e.g. ".917 y ^{xx} ".
040	R/S	84			EEX	43	
	Sto 1	33 01	Key 8		8	08	
	.9	09			X	71	
	R/S	84			g frac.	32 83	
	Sto $\emptyset$	33 00	Key 9	100	EEX	43	
	rcl B	34 12			2	02	
	Sto + i	33 61 24	modifies [i]		X	71	
	$\emptyset$	00			f int.	31 83	
	h St I	35 33			• FLBL $\emptyset$	31 25 00	
	Sto B	33 12			rcl A	34 11	
050	f. w/data	31 41	optional		h $x-y$	35 52	
	R/S	84			g $x \leq y$	32 71	
	• FLBL D	31 25 14	Decoding start		Sto 1	22 01	
	h CF $\emptyset$	35 61 00			h $x-y$	35 52	
	• g lll d	32 25 14		110	—	51	
	Sto E	33 15			Sto $\emptyset$	22 00	
	h CF 3	35 61 03			• FLBL 1	31 25 01	

REGISTERS

0 Key 9	1 Key 8	2 Key 7	3 Key 6	4 Key 5	5 Key 4	6 Key 3	7 Key 2	8 Key 1	9 Modifier
S0 modifier derived fr. prev. entry	S1	S2	S3	S4	S5	S6	S7	S8	S9
A Number of Alphabet Charact.	B initially Key " $\emptyset$ ", later empty.	C constant	D	E prev. entry	F counter				

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
113	f ISZ	31 34			f CL REG	31 43	
	rcl E	34 15		170	f P-S	31 42	"PANIC BUTTON" Sequence.
	h x-y	35 52			f CL REG	31 43	
	h F? 2	35 71 02			cl x	44	
	h RTN	35 22			ent	41	
	—	51			ent	41	
	• f LBL 3	31 25 03			ent	41	
120	rcl A	34 11			+	61	
	disp 0	23 00			displ 2	23 02	
	h x-y	35 52			f FIX	31 23	
	g x ≤ y	32 71			h CF 3	35 61 03	
	gto 2	22 02		180	h CF 2	35 61 02	
	h x-y	35 52			h CF 1	35 61 01	
	—	51			h CF 0	35 61 00	
	gto 3	22 03			R/S	84	
	• f LBL 2	31 25 02			• f LBL C	31 25 13	Count routine
	f x < 0	31 71			h x-I	35 24	
130	f GSB 4	31 22 04			h PAUSE	35 72	
	f x = 0	31 51			h x-I	35 24	
	f GSB 4	31 22 04			gto 7	22 07	
	• f LBL 7	31 25 07			• f LBL 8	31 25 08	part of "random" # generator routine.
	h PAUSE	35 72	See p.2, Block IV.	190	f LOG	31 53	
	h F? 3	35 71 03			g frac.	32 83	
	gto 6	22 06			sto + 9	33 61 09	
	gto 7	22 07			rcl 9	34 09	
	• f LBL 6	31 25 06			g frac.	32 83	
	h F? 0	35 71 00			sto 9	33 09	
140	gto 5	22 05			h last x	35 82	
	h R↓	35 53			h RTN	35 22	
	f ln	31 52	used for decoding only		• g LBL c	32 25 13	figure check routine (shows number just entered)
	g frac.	32 83			rcl E	34 15	
	sto + 9	33 61 09		200	h Pause	35 72	
	h R↑	35 54			h R↓	35 53	
	gto f b	22 31 12			gto 7	22 07	
	• f LBL 4	31 25 04			• f LBL 5	31 25 05	used for encoding only
	rcl A	34 11			sto B	33 12	
	+	61			f P-S	31 42	
150	f x < 0	31 71			sto + 0	33 61 00	
	gto 2	22 02			rcl 0	34 00	
	h RTN	35 22			rcl B	34 12	
	• f LBL E	31 25 15	Encoding Start		—	51	
	f P-S	31 42		210	f ln	31 52	
	sto 0	33 00			g frac.	32 83	
	f P-S	31 42			f P-S	31 42	
	g LBL e	32 25 15			sto + 9	33 61 09	
	h SF 0	35 51 00			rcl B	34 12	
	h SF 2	35 51 02			f P-S	31 42	
160	g GSB 2	32 22 14			sto 0	33 00	
	h SF 2	35 51 02			f P-S	31 42	
	+	61			gto f &	22 31 12	
	gto 3	22 03		220			
	• g LBL b	32 25 12					
	h F? 2	35 71 02	decision as to en- or decoding.				
	gto f e	22 31 15					
	gto D	22 14					
	• f LBL B						

LABELS					FLAGS	SET STATUS		
A START	Blank out everything	Count	Decode	Encode	0 used	FLAGS	TRIG	DISP
a	b used	check of prev. entry	d used	e used	1	ON OFF		
0 used	1 used	2 used	3 used	4 used	2 used	0 ☐ ☒	DEG ☒	FIX ☒
5 used	6 used	7 used	8 used	9	3 used	1 ☐ ☒	GRAD ☐	SCI ☐
						2 ☐ ☒	RAD ☐	ENG ☐
						3 ☐ ☒		n ☒