01829D

Program Description I

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 Ø 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 Ø 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

Program Description II

I

Sketch(es)

SAMPLE ALPHABET

(	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	.	K
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		O	7	Y	4	Ø	Z	8	P	Q	5	R	Z	?	S	T	B	U	V	C	-	A	R	×	W
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	×	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						

II

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

a→			E	X	A	M	P	L	E		O	F		C	O	M	P	U	T	E	R	-	G	E	N	E	R
b→	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
c→	33	89	28	18	80	18	79	56	02	15	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	54	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)
99.74285369

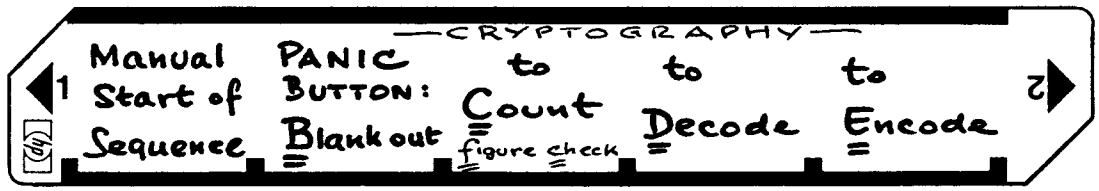
Key	
1	3.852 964 712
2	5.471 892 365
3	7.829 531 645
4	6.142 798 653
5	4.517 965 329
6	2.718 469 325
7	1.728 953 645
8	9.100 578 425
9	7.250 840 935

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.)

User Instructions I



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Agree upon the "Alphabet" that will	[p.2, SKETCH]		
	form the basis of the coding process.			
	Go to 2.			
2	Agree upon the "Initiator" and the	[p.2, BLOCK 3]		
	9 "Keys". Go to 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 / optional	[any]	[Ø] / Ø.
5	Run in Data card side 1. Go to 6.			
6	"Translate" the message/text into	[see p.2,		
	numerals in accordance with the Al-	lines a		
	phabet mentioned in Step 1. Go to 7.	and b]		
7	For encoding, go to 8; decoding to 12.			
8	Take the first translated character of	[p.2, line b]		
	the text (whether it previously was letter,			
	numeral, space, punctuation, whatever) and			
	key that number in. Press E (for Enco-			
	de). Go to 9.	≤ [A], pos.	E	...
9	The new number showing during the			... rep'g
	repeating Pause is the encoded num-	[see p.2,		
	ber equivalent of the text's original	line c]		
	character. See Note 3! Write it down			
	(to start/continue the string of code			
	numbers that result from continuing en-			
	coding). Go to 10 if continuing; else go			
	to Step 11.			

User Instructions II

CRYPTO DATA CARD (side 1 only)

- optional -

1 ◀ 2 ▶

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	∅.
12	Key in 1st 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:		f c	. . .
	To verify "serial" number of the number now showing, press "Count":		C	. . .
	Continue only after clear or code number shows again.			

Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	• f LBL A	31 25 11	
	h CF 2	35 61 02	
	Sto A	33 11	
	g Frac.	32 83	
	Sto B	33 12	
	1	01	
	Ø	00	
	X	71	
	f int.	31 83	
010	9	09	
	h x-y	35 52	
	g x=y	32 51	
	3	03	Can be any number from Ø thru 8.
	h x-I	35 24	
	Rcl A	34 11	
	f int	31 83	
	Sto A	33 11	number of Alph. Char.
	1	01	
	R/S	84	
020	Sto 8	33 08	Key 1
	2	02	
	R/S	84	
	Sto 7	33 07	Key 2
	3	03	
	R/S	84	
	Sto 6	33 06	Key 3
	4	04	
	R/S	84	
	Sto 5	33 05	Key 4
030	5	05	
	R/S	84	
	Sto 4	33 04	Key 5
	6	06	
	R/S	84	
	Sto 3	33 03	Key 6
	7	07	
	R/S	84	
	Sto 2	33 02	Key 7
	8	08	
040	R/S	84	
	Sto 1	33 01	Key 8
	9	09	
	R/S	84	
	Sto Ø	33 00	Key 9
	rcl B	34 12	
	Sto + i	33 61 24	modifies [i]
	Ø	00	
	h St I	35 33	
	Sto B	33 12	
050	f: w/data	31 41	optional
	R/S	84	
•	f LBL D	31 25 14	Decoding Start
	h CF Ø	35 61 00	
•	g LBL d	32 25 14	
	Sto E	33 15	
	h CF 3	35 61 03	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	EEX	43	
	3	03	
	h π	35 73	
060	—	51	
	g frac.	32 83	
	Sto C	33 13	
	rcl 8	34 08	
	f GSB 8	31 22 08	
	Sto + 8	33 61 08	
	rcl 7	34 07	
	f GSB 8	31 22 08	
	Sto + 7	33 61 07	
	rcl 6	34 06	
070	f GSB 8	31 22 08	
	Sto + 6	33 61 06	
	rcl 5	34 05	
	f GSB 8	31 22 08	
	Sto + 5	33 61 05	
	rcl 4	34 04	
	f GSB 8	31 22 08	
	Sto + 4	33 61 04	
	rcl 3	34 03	
	f GSB 8	31 22 08	
080	Sto + 3	33 61 03	
	rcl 2	34 02	
	f GSB 8	31 22 08	
	Sto + 2	33 61 02	
	rcl 1	34 01	
	f GSB 8	31 22 08	
	Sto + 1	33 61 01	
	rcl 0	34 00	
	f GSB 8	31 22 08	
	Sto + 0	33 61 00	
090	rcl 9	34 09	
	h absol.	35 64	
	rcl C	34 13	
	X	71	
	g frac.	32 83	
	f $\sqrt{x}$	31 54	can be any fractional power, e.g. ".917 y^x".
	EEX	43	
	8	08	
	X	71	
	g frac.	32 83	
100	EEX	43	
	2	02	
	X	71	
	f int.	31 83	
•	f LBL Ø	31 25 00	
	rcl A	34 11	
	h x-y	35 52	
	g x≤y	32 71	
	gto 1	22 01	
	h x-y	35 52	
110	—	51	
	gto Ø	22 00	
•	f LBL 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 "Ø"; later empty.	C constant	D	E prev. entry	I counter

Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	f ISZ	31 34	
	rce E	34 15	
	h x-y	35 52	
	h F? 2	35 71 02	
	h RTN	35 22	
	—	51	
•	f LBL 3	31 25 03	
120	rcl A	34 11	
	disp ø	23 00	
	h x-y	35 52	
	g x≤y	32 71	
	gto 2	22 02	
	h x-y	35 52	
	—	51	
	gto 3	22 03	
•	f LBL 2	31 25 02	
	f x<ø	31 71	
130	f GSB 4	31 22 04	
	f x=ø	31 51	
	f GSB 4	31 22 04	
•	f LBL 7	31 25 07	
	h PAUSE	35 72	See p.2, Block IV.
	h F? 3	35 71 03	
	gto 6	22 06	
	gto 7	22 07	
•	f LBL 6	31 25 06	
	h F? ø	35 71 00	
140	gto 5	22 05	
	h R↓	35 53	Used for decoding only
	f ln	31 52	
	g frac.	32 83	
	Sto + 9	33 61 09	
	h R↑	35 54	
	gto f b	22 31 12	
•	f LBL 4	31 25 04	
	rcl A	34 11	
	+	61	
150	f x<ø	31 71	
	gto 2	22 02	
	h RTN	35 22	
•	f LBL E	31 25 15	Encoding Start
	f P-S	31 42	
	Sto ø	33 00	
	f P-S	31 42	
	g LBL e	32 25 15	
	h SF ø	35 51 00	
	h SF 2	35 51 02	
160	g GSB d	32 22 14	
	h SF 2	35 51 02	
	+	61	
	gto 3	22 03	
•	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		

STEP	KEY ENTRY	KEY CODE	COMMENTS
	f CL REG	31 43	"PANIC BUTTON" Sequence.
170	f P-S	31 42	
	f CL REG	31 43	
	cl x	44	
	ent	41	
	ent	41	
	ent	41	
	+	61	
	disp. 2	23 02	
	f FIX	31 23	
	h CF 3	35 61 03	
180	h CF 2	35 61 02	
	h CF 1	35 61 01	
	h CF ø	35 61 00	
	R/S	84	
•	f LBL C	31 25 13	Count routine
	h x-I	35 24	
	h PAUSE	35 72	
	h x-I	35 24	
	gto 7	22 07	
•	f LBL 8	31 25 08	part of "random" # generator routine.
190	f LOG	31 53	
	g frac.	32 83	
	Sto + 9	33 61 09	
	rce 9	34 09	
	g frac.	32 83	
	Sto 9	33 09	
	h last x	35 82	
	h RTN	35 22	
•	g LBL c	32 25 13	figure check routine (shows number just entered)
	rcl E	34 15	
200	h Pause	35 72	
	h R↓	35 53	
	gto 7	22 07	
•	f LBL 5	31 25 05	Used for encoding only
	Sto B	33 12	
	f P-S	31 42	
	Sto + ø	33 61 00	
	rcl ø	34 00	
	rcl B	34 12	
	—	51	
210	f ln	31 52	
	g frac.	32 83	
	f P-S	31 42	
	Sto + 9	33 61 09	
	rcl B	34 12	
	f P-S	31 42	
	Sto ø	33 00	
	f P-S	31 42	
	gto f B	22 31 12	
220			

LABELS

A START	Blank out everything	Count	Decode	Encode	0 Used
a	b Used	check of prev. entry	d Used	e Used	1
0 Used	1 Used	2 Used	3 Used	4 Used	2 Used
5 Used	6 Used	7 Used	8 Used	9	3 Used

SET STATUS

FLAGS		TRIG	DISP
	ON OFF		
0	☐ ☒	DEG ☒	FIX ☒
1	☐ ☒	GRAD ☐	SCI ☐
2	☐ ☒	RAD ☐	ENG ☐
3	☐ ☒		n ø