Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers

https://youtu.be/Yi0-iVxmRIU

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers

- 딥러닝 네트워크를 활용하여 평문 및 암호문쌍으로 키 값을 찾아내는 공격
- S-DES, SPECK, SIMON에 대해 수행
  - S-DES

    성공 (비트키, 텍스트키)
  - SPECK, SIMON

    실패 (비트키)

    성공(텍스트키)

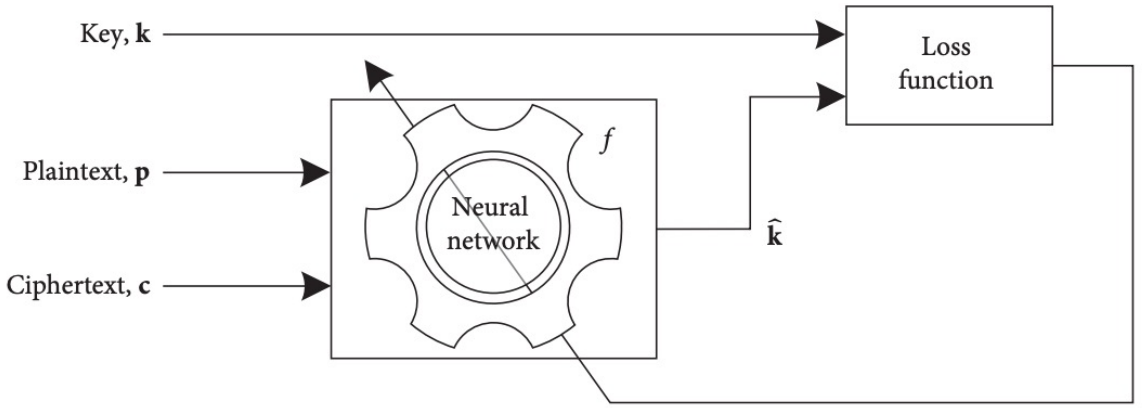
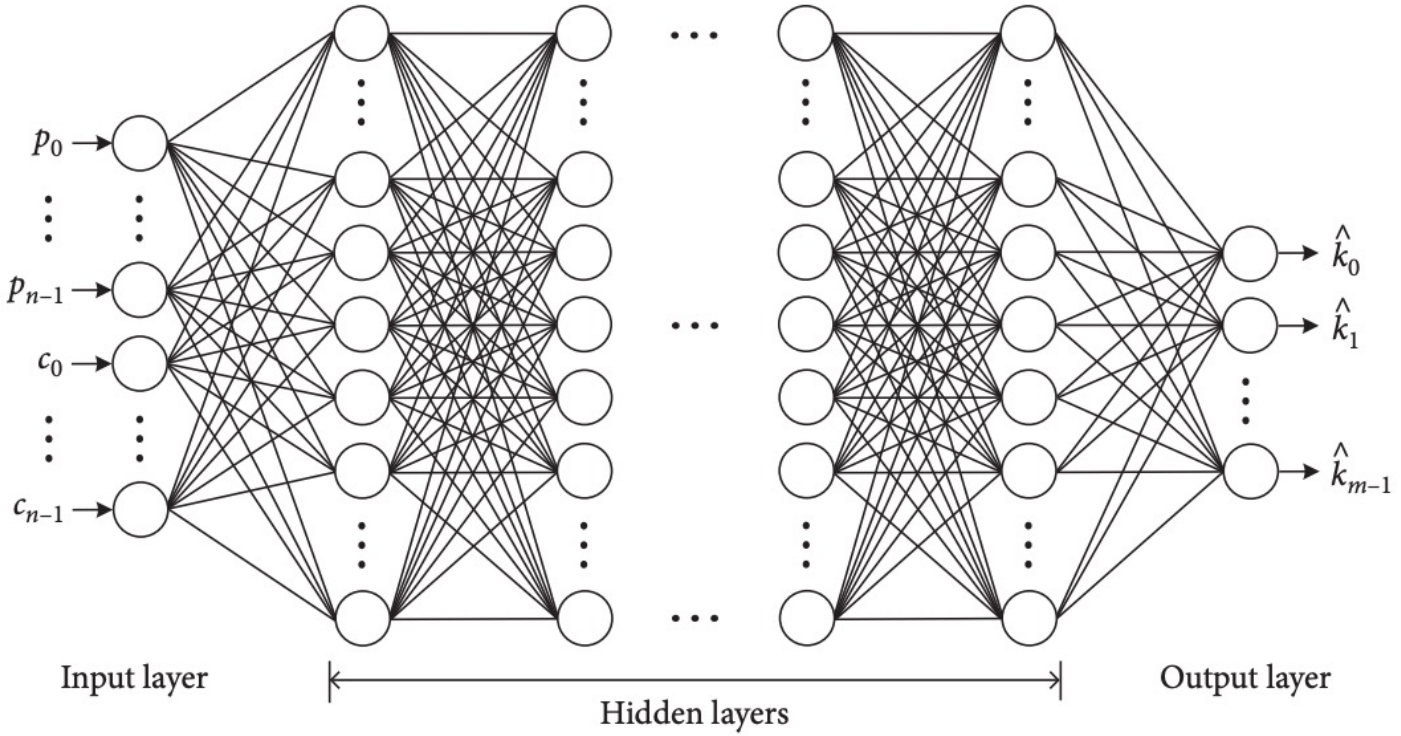


FIGURE 1: A schematic diagram of the DL-based cryptanalysis.

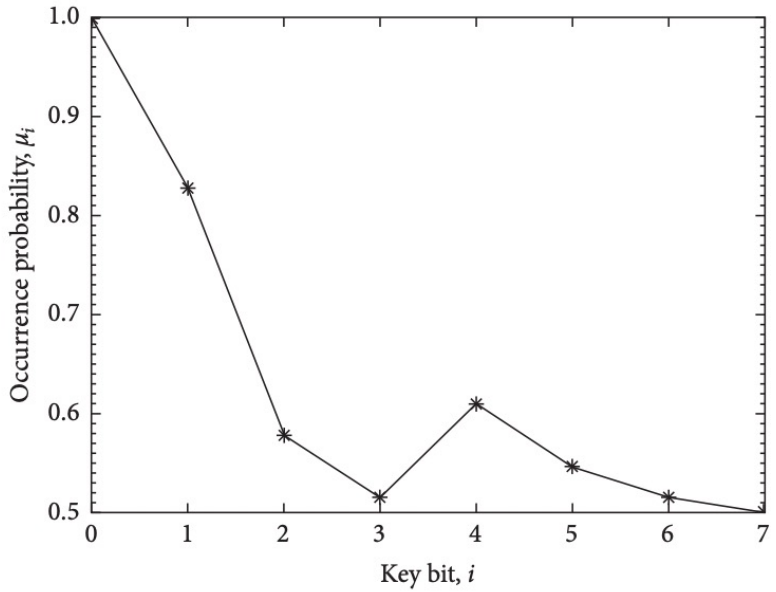
Model architecture

- 입력 평문 및 암호문 쌍의 각 비트를 각 뉴런에 할당 (즉, n비트의 평문-암호문인 경우, $2n$개의 입력 뉴런)
- 키의 각 비트를 출력 뉴런에 할당 (n 비트의 키 사용 시, n 개의 출력 뉴런)
- 각 뉴런의 값 그대로 나오도록 relu 활성화 함수 적용
- MSE 손실 함수 사용
- 하이퍼 파라미터 최적화 방식은 그리드 서치를 사용한듯
  (히든 뉴런 128, 256, 512 / 레이어 수 3, 5, 7 의 모든 경우의 수에 대해 성능 평가한 후 가장 좋은 조합을 사용)



Dataset

- **비트키**는 공격 정확도가 높지 않고,
  SPECK과 SIMON에서는 실패했기 때문에 **텍스트키** 사용
- 텍스트키는 A~Z, a~z, 0~9, !, @ : **총 64개**
  S-DES는 8-bit 평문 및 10-bit 키 사용
  → **랜덤 비트 평문-암호문** (8-bit) 및 **텍스트 + 2 비트의 키**(10-bit)
  → 실제 키 공간은 2^6이므로 6-bit 랜덤키와 비슷
  → 그러나 이 경우도, 해당 텍스트만 사용할 경우
  각 비트의 발생 확률이 다름



1번째 비트는 아예 0으로 고정
2번째 비트는 높은 확률로 1

이러한 특성으로 인해
예측하기 더 쉬울 것으로 생각

BIN	HEX	Char.
00000000	0	NUL
00000001	1	SOH
00000010	2	STX
00000011	3	ETX
00000100	4	EOT
00000101	5	ENQ
00000110	6	ACK
00000111	7	BEL
00001000	8	BS
00001001	9	HT
00001010	0A	LF
00001011	0B	VT
00001100	0C	FF
00001101	0D	CR
00001110	0E	SO
00001111	0F	SI
00010000	10	DLE
00010001	11	DC1
00010010	12	DC2
00010011	3	DC3
00010100	14	DC4
00010101	15	NAK
00010110	16	SYN
00010111	17	ETB
00011000	18	CAN
00011001	19	EM
00011010	1A	SUB
00011011	1B	ESC
00011100	1C	FS
00011101	1D	GS
00011110	1E	RS
00011111	1F	US
00100000	20	SPACE
00100001	21	!
00100010	22	"
00100011	23	#
00100100	24	$
00100101	25	%
00100110	26	&
00100111	27	'
00101000	28	(
00101001	29	)
00001010	2A	*

BIN	HEX	Char.
00101011	2B	+
00101100	2C	,
00101101	2D	-
00101110	2E	.
00101111	2F	/
00110000	30	0
00000110	31	1
00110010	23	2
00110011	33	3
00110100	34	4
00110101	35	5
00110110	36	6
00001100	37	7
00111000	38	8
00111000	39	9
00111010	3A	:
00111011	3B	;
00111100	3C	<
00111101	3D	=
00111110	3E	>
00111111	3F	?
01000000	40	@
01000001	41	A
01000010	42	B
01000011	43	C
01000100	44	D
01000101	45	E
01000110	46	F
01000111	47C	G
01001000	48	H
01001001	49	I
01001010	4A	J
01001011	4B	K
01001100	4C	L
01001101	4D	M
01001110	4E	N
01001111	4F	O
01010000	50	P
01010001	51	Q
01010010	52	R
01010011	53	S
01010100	54	)
01010101	55	T

BIN	HEX	Char.
01010110	56	V
01010111	57	W
01011000	58	X
01011001	59	Y
01011010	5A	Z
01011011	5B	[
01011100	5C	₩
01011101	5D	]
01011110	5E	^
01011111	5F	–
01100000	60	`
01100001	61	a
01100010	62	b
01100011	63	c
01100100	64	d
01100101	65	e
01100110	66	f
01100111	67	g
01101000	68	h
01101001	69	i
01101010	6A	j
01101011	6B	k
01101100	6C	l
01101101	6D	m
01101110	6E	n
01101111	6F	o
01110000	70	p
01110001	71	q
01110010	72	r
01110011	73	s
01110100	74	t
01110101	75	u
01110110	76	v
01110111	77	w
01111000	78	x
00100011	79	y
01111010	7A	z
01111011	7B	{
01111100	7C	|
01111101	7D	}
01111110	7E	~
01111111	7F	DEL

Dataset

- 데이터&모델의 세부사항 및 훈련/검증 데이터 개수

	S-DES	Simon	Speck
Block size	8	32	32
Key size	10	64	64
Round	2	32	22
Epoch	5000		
layers	512 units, 5 hidden layers		
The number of data	5만/1만	50만/ 100만	

평가 지표

- **Bit Accuracy Probability(BAP)**
  : 전체 데이터에 대한 예측 성공 확률
  → 1번째 비트가 100개의 데이터 중에서 80개를 예측 성공했다면, 0.8

$$\rho_i = \frac{1}{N_s} \sum_{j=1}^{N_s} \mathrm{XNOR}\left(\mathbf{k}_i^{(j)}, \tilde{\mathbf{k}}_i^{(j)}\right),$$

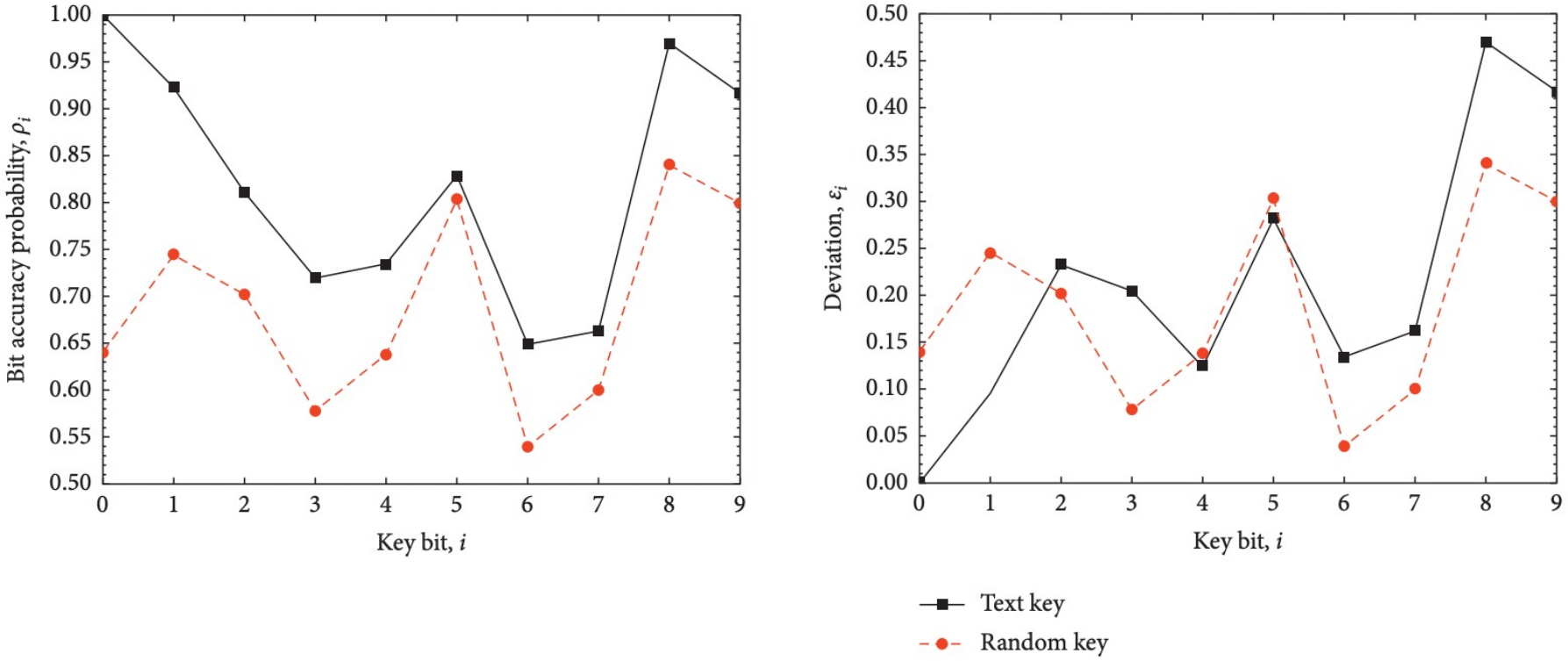
- **Deviation**
  : BAP - 키 발생 확률 ; 실제 키 발생 빈도가 다르므로 제거하고 판단하기 위함

$$\varepsilon_i = \rho_i - \mu_i,$$

해당 값이 양수이면, 공격 성공 (예측 확률이 발생 확률보다 낮을 경우 공격 실패)

실험 결과

- **S-DES**
  - 1번째 비트 1.00 → 원래 발생 확률 1로 값이 0인 비트
  - 비트보다 텍스트 키의 경우 더 잘 예측해냄
  - **BAP**를 보면, 1, 5, 8번째 비트는 암호 분석에 취약, **6번째 비트는 안전**
  - **Deviation**을 보면, **2,5,8 (텍스트 키), 1,5,8(비트키)는 공격에 취약**
    (i번째 키비트에 대한 예측 확률이 낮아야 deviation이 낮아짐)



실험 결과

- **S-DES**
  동일한 키로 암호화 된 M개의 평문-암호문 쌍이라고 할 경우, (dataset 개수 아님)
  모든 키 비트에 대해 특정 성공확률을 달성하기 위해 필요한 최소 M (다음 수식에 의해 계산)

  - **비트 키**
    BAP의 최소 값이 0.5389
    → 0.9의 성공 확률을 얻기 위해 M = 271, 0.99는 M=891
  - **텍스트 키**
    BAP의 최소값이 0.6484
    → 0.9의 성공 확률 얻기 위해 M = 19, 0.99는 M = 59

$$\alpha_i(M) = \Pr\left(X \geq \frac{M}{2} + 1\right) = 1 - \Pr\left(X \leq \frac{M}{2}\right)$$

$$= 1 - \sum_{j=0}^{M/2} \binom{M}{j} \rho_i^j (1-\rho_i)^{M-j}.$$

$$M_i^* = \min\left\{M \mid \alpha_i(M) \geq \tau\right\}.$$

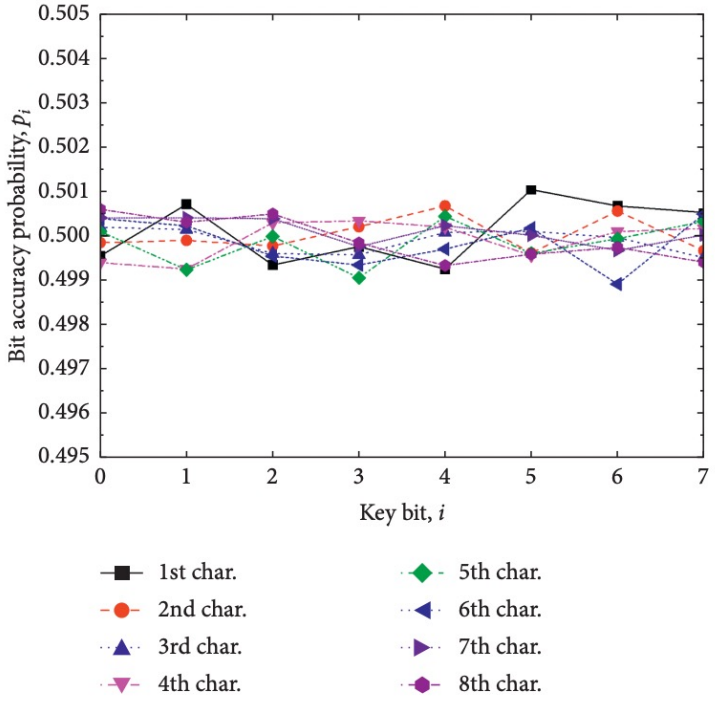
*i번째 비트에 대한 BAP를 p로 가지는 이항 분포를 나타냄
M개 중 절반 이상 성공할 경우에 대한 확률 = 1-(0개 맞출 확률 + … + M/2개 맞출 확률)
즉 α_i 는 절반 이상을 맞출 수 있는 확률이 특정 성공 확률 보다 커질 때의 최소값
동일 키로 암호화된 평문-암호문 쌍이 M이 271개면 137개 이상을 맞출 수 있으며, 0.9의 성공확률을 가진다고 이해..

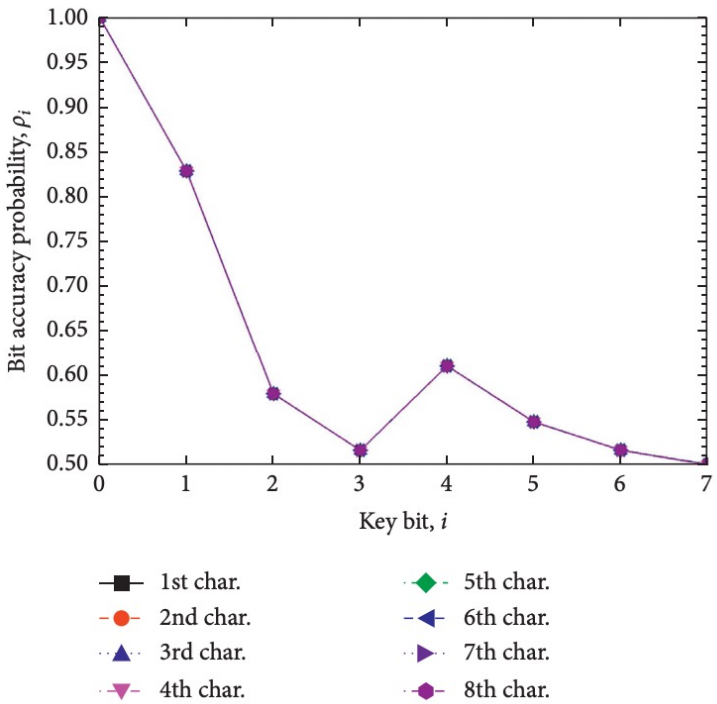
실험 결과

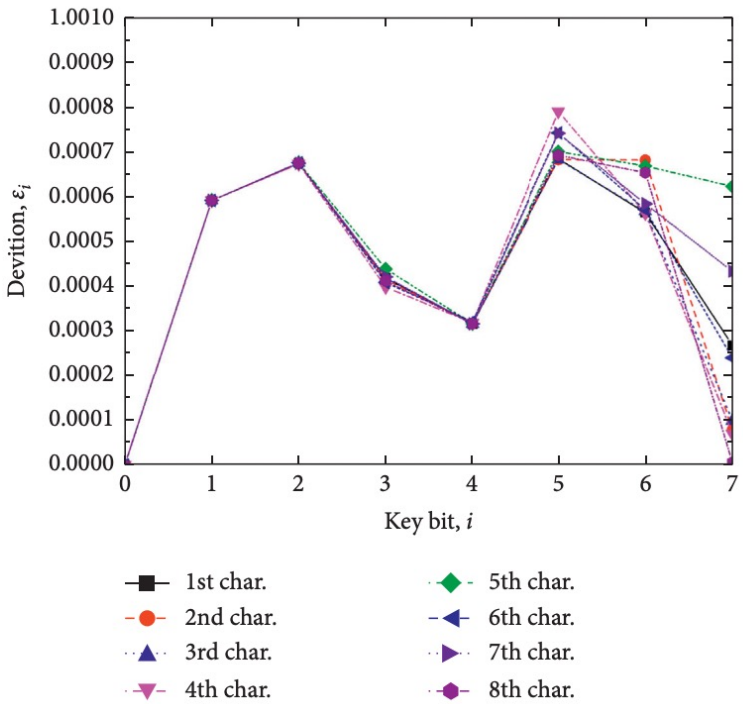
- **Simon , Speck**
  - 64bit 키 → 텍스트키 8개 → **실제 키 공간은 2^{48}**
  - full-round speck, simon에 대한 암호분석 연구 결과 X
  - 이전 연구들에서 24라운드 simon → 2^{63}에서 2^{32}로 감소
    14라운드 speck → 2^{63}에서 2^{31}로 감소

실험 결과

- **Simon**
  - 비트키의 경우 실패 (BAP 평균이 거의 0.5→ 거의 예측하지 못함)
  - 텍스트키의 경우 성공
    - K3에서 BAP가 0.51603로 가장 낮음
      그러나 이때도 발생 확률보다 0.0004 크기때문에 (deviation이 양수) 공격 성공했다고 함
    - M = $2^{10.58}$이면 0.9의 확률, M = $2^{12.34}$이면 0.99의 확률로 키를 찾을 수 있음

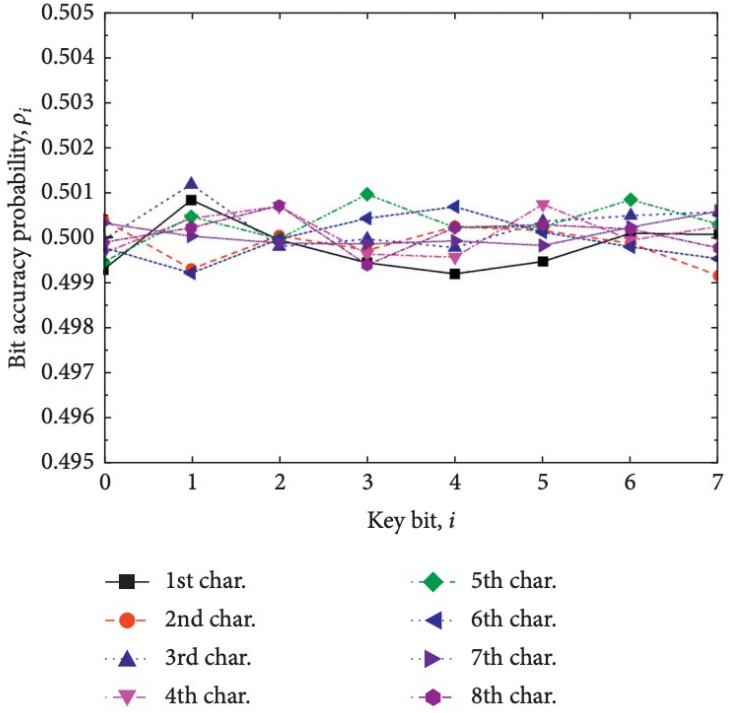


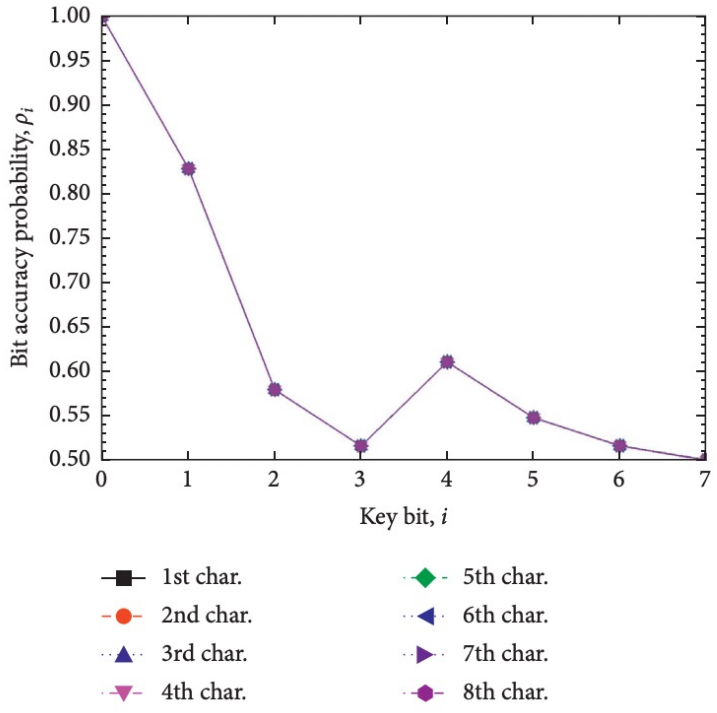


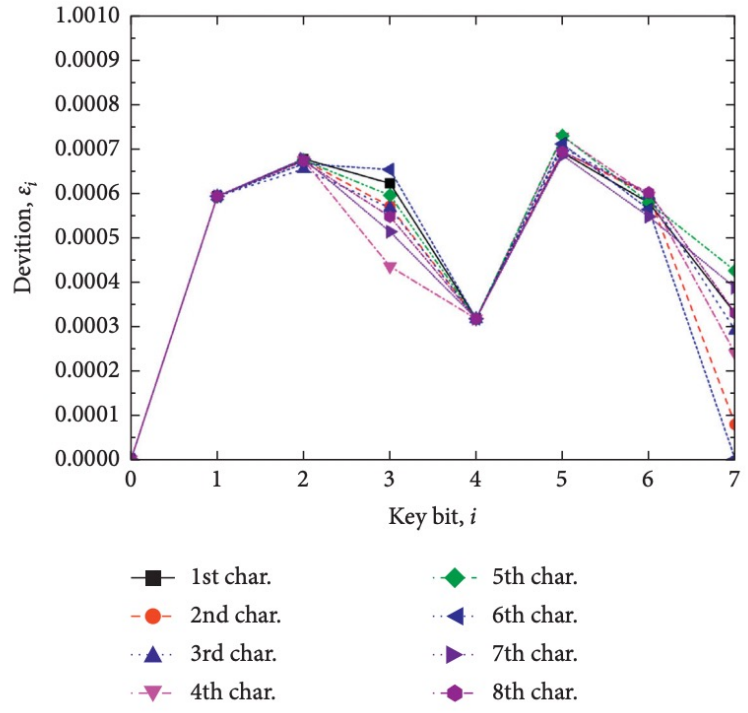


실험 결과

- **Speck**
  - 비트키의 경우 실패 (BAP 평균이 거의 0.5→ 거의 예측하지 못함)
  - 텍스트키의 경우 성공
    - k3에서 BAP가 0.51607로 가장 낮지만, 이는 발생 확률보다 0.00044 큰 값이므로 성공
    - M = $2^{10.57}$이면 0.9의 확률, M = $2^{12.33}$이면 0.99의 확률로 키를 찾을 수 있음







감사합니다.