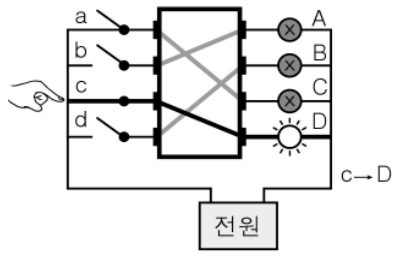
◆ 12년 3월 고3 19~21번

[19~21] 다음 글을 읽고 물음에 답하시오.

암호를 만드는 방식에는 크게 전치(轉置) 방식과 환자(換字) 방식이 있다. 전치 방식은 전하고자 하는 메시지인 평문의 철자 위치를 바꾸어 암호문을 만드는 방식이고, 환자 방식은 평문 철자를 일정한 규칙에 따라 다른 문자로 바꾸어 암호문을 만드는 방식이다. 더 복잡하게 하고자 할 때는 전치 방식과 환자 방식을 함께 적용한 혼합 방식을 사용할 수도 있다. 이렇게 평문을 암호문으로 만드는 암호화의 규칙을 '알고리즘(algorism)'이라 하고 역으로 암호문을 평문으로 푸는 해독의 규칙을 '키(key)'라 한다.

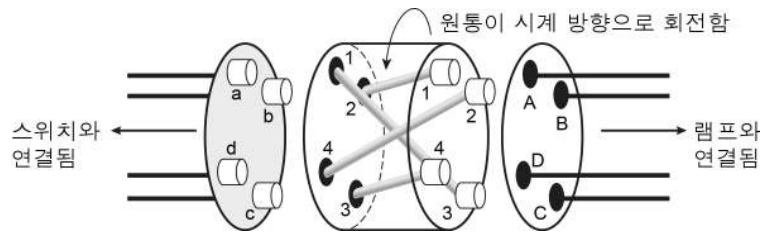
암호의 원리는 현대 문명의 발달과 함께 다양한 암호 장치들로 구현되었다. 특히 전신(電信)에 의한 송·수신이 가능하게 되면서 전기 암호 장치가 개발되었다.



[그림 1] 전기 암호 장치의 구조

[그림 1]에서 스위치 a, b, c, d와 연결된 램프 A, B, C, D로 구성된 이 장치는, 스위치 역할을 하는 키보드에서 평문 철자를 입력했을 때 불이 켜지는 램프가 암호 철자를 나타낸다. [그림 1]에서처럼 송신자가 평문 철자 c를 입력했을 때 c에 연결된 램프 D가 켜지고 이 암호 철자 D를 전신으로 받은 수신자는 암호 철자인 D를 눌러 D에 연결된 원래의 평문 철자 c를 얻게 되는 방식이다.

그러나 전기 암호 장치는 회로 구조가 파악되면 쉽게 암호가 노출될 수 있고, 그런 경우 다시 회로 구조를 바꿔야 하는 불편함이 있었기 때문에 전시(戰時)와 같이 분초를 다투는 위급한 상황에는 적합하지 않았다. 이런 문제를 극복하기 위해 회전하는 원통 속에 전기 회로를 넣은 '에니그마'가 개발되었다.



[그림 2] 에니그마의 구조

에니그마는 원통이 회전함에 따라 스위치와 램프의 연결 상태가 계속 바뀌기 때문에 회로 구조를 조작하지 않고도 평문 철자와 암호 철자의 대응 규칙을 쉽게 바꿀 수 있다. 예를 들어 원통이 회전하지 않는 경우에는 평문 'b'를 입력하면 [그림 2]의 회로 구조를 거쳐 암호문 'A'로 나타나지만(b-2-1-A), 원통이 시계 방향으로 1/4바퀴 돌아간 경우에는 같은 'b'를 입력하더라도 원통 속의 회로가 다른 단자와 연결되면서 암호문 'C'로 나타나게 된다(b-3-4-C). 따라서 암호를 파악하기 위해서는 원통 바퀴 내부의 회로 구조뿐만 아니라 한 글자를 입력할 때마다 원통이 언제, 얼마나 회전하는지 등 원통의 회전 규칙도 함께 알아야 한다.

19. '전기 암호 장치'와 '에니그마'에 대한 설명으로 적절하지 <u>않은</u> 것은?

① 전기 암호 장치의 한계를 극복하기 위해 에니그마를 개발했다.
② 전기 암호 장치와 에니그마는 전기 회로를 통해 환자 방식을 구현했다.
③ 전기 암호 장치와 에니그마의 스위치는 평문 철자, 램프는 암호 철자를 나타낸다.
④ 에니그마는 전기 암호 장치와 같이 회로 구조만 알아도 암호를 해독할 수 있다.
⑤ 전기 암호 장치는 에니그마와 달리 대응 규칙을 쉽게 바꾸기 어렵다는 문제점이 있다.

20. 위 글을 통해 <보기>를 이해한 것으로 적절하지 <u>않은</u> 것은?

<보 기>

영희가 철수에게 보낸 문자　　철수가 영희에게 보낸 문자

① '널 사랑해'는 영희가 전하고 싶은 평문이군.
② '널 사랑해'를 '해랑사널'로 바꾼 것은 전치 방식이군.
③ '해랑사널'을 'gofkdtksjf'로 바꾼 것은 환자 방식이군.
④ '한글 입력 모드에서 이 영문을 입력하고, 그걸 거꾸로 읽어'는 알고리즘에 해당하는군.
⑤ 철수가 보낸 '해랑사도나'는 전치 방식으로 보낸 것이군.

21. 위 글의 내용을 바탕으로 할 때, <보기>의 괄호 안에 들어갈 '암호 철자'는?

<보 기>

[그림 2]의 에니그마로 암호를 보내고자 한다. 에니그마에 첫 번째 평문 철자를 입력할 때는 원통이 고정되어 있고, 그 이후에는 평문 철자를 입력할 때마다 원통이 반시계 방향으로 1/4만큼 회전하도록 설계되었다고 하자. 평문 철자 'bc'를 입력했을 때 나올 암호 철자는 (　　)이다.

① AB　② AC　③ BC　④ DA　⑤ DB

- 출전: 루돌프 키펜한 저. 《암호의 세계》
- 정답: 19.④ 20.④ 21.①

◆ 07 경찰대 1차 47~50번

[47-50] 다음 글을 읽고 물음에 답하시오.

1941년 6월 22일 독일은 소련을 침공했다. 소련은 독일뿐 아니라 일본까지도 동시에 막아야 할 위기에 처했다. 그 때 동경으로부터 날아온 비밀 정보는 전쟁의 흐름을 바꾸어 놓았다. "일본은 소련을 공격할 계획이 없음." 마샬 추코프 사령관은 이 보고에 따라 시베리아 주둔 몇 개 사단과 탱크, 비행기를 모스크바 전선으로 이동시켜 독일군을 저지했다.

소련은 어떤 방법으로 일본에게 들키지 않고 비밀 정보를 송신할 수 있었을까? 스파이 클라우젠이 사용한 방법은 다음과 같다. 먼저 오른쪽의 암호 표에서처럼 맨 위에 **subway**를 쓴 다음, 그 밑에 나머지 알파벳을 순서대로 적고, 마지막에는 마침표와 빗금을 첨가했다. 그리고 빈도가 높은 8개 철자 'a, s, i, n, t, o, e, r'에는 0부터 7까지 숫자를 매기는데, 왼쪽 칸의 's'에서부터 아래로 내려가면서 숫자를 적었다. 나머지 철자들도 같은 방법으로 해서 80에서 99까지의 수를 기입했다. 이것으로 정보 문서는 숫자로 변환되었다. 가령, 클라우젠이 '**no attack**(공격 없음)'이라는 정보를 암호화 한다고 가정해 보자. 그러면 위의 방법에 따라 '**no attack**'은 **729456658088**로 바뀐다.

s	u	b	w	a	y
c	d	e	f	g	h
i	j	k	l	m	n
o	p	q	r	t	v
x	z	.	/		

s	u	b	w	a	y
0				5	
c	d	e	f	g	h
		3			
i	j	k	l	m	n
1					7
o	p	q	r	t	v
2			4	6	
x	z	.	/		

s	u	b	w	a	y
0	82	87	91	5	97
c	d	e	f	g	h
80	83	3	92	95	98
i	j	k	l	m	n
1	84	88	93	96	7
o	p	q	r	t	v
2	85	89	4	6	99
x	z	.	/		
81	86	90	94		

그런데 ㉠이런 방법으로만 암호화를 하면 일본 정보부가 쉽게 알아챌 것이다. 그래서 그는 모스크바와 미리 약속한 방식대로 이 숫자들에다 <독일 제국 연감>(1935년)에서 선택한 다른 숫자들을 결합하였다. 가령, 연감의 34째 면, 23째 줄, 6째 단에서 '4230, 5166, 7821, 9421'을 선택한 다음, 이것을 세 번째 숫자부터 사용했다. 이렇게 만들어진 수 '**305166782194……**'가 해독의 열쇠이다. 그는 이것을 앞에서 만든 암호문 **729456658088** 아래에 배열했다.

729456658088
305166782194

그 다음에는 두 줄의 수를 위아래로 더하되, 9를 넘더라도 10을 윗자리로 넘기지 않는 방법을 사용했다. 즉, '7+3'은 10이 아니라 0이다. 이렇게 해서 나온 결과가 '중간 암호문'이다. ㉡여기에 다시 연감의 해당 위치를 나타낸 수 **34236**을 덧붙여야만 했다. 이것은 34면 23째줄 6째 단을 뜻한다. 덧붙이는 방법은, 먼저 이 수를 중간 암호문의 앞자리 5개와 더한 다음에(이때에도 10을 윗자리로 넘기지 않고 더했음), 그 결과를 중간 암호문의 앞에 배열하는 것이다. 이리하여 완성된 암호문을 다섯 자리씩 간격을 두어 나열하면, '**no attack**'은 다음과 같은 암호로 바뀐다. (㉢) **02451 23301 72.**

일본 정보부에서는 클라우젠이 만든 암호문들을 전쟁이 끝나도록 풀지 못했다. 사용된 암호화 과정이 워낙 정교한데다 평범한 책을 활용했기 때문이다. 클라우젠의 가택 수색에서도 별다른 주위를 끌지 못했던 연감이 실마리였다는 사실을 일본은 눈치 채지 못했다.

47. 다음 중 모스크바와 클라우젠 사이에 암호 해독을 위하여 사전에 약속된 것으로 보기 어려운 것은?

① 연감에서 선택한 숫자의 위치
② 1935년 <독일 제국 연감>을 활용
③ 빈도가 높은 알파벳으로 선택된 것들
④ 암호문과 해독의 열쇠를 더하는 방법
⑤ 빈도가 높지 않은 알파벳에 숫자를 매기는 순서

48. 다음 중 ㉠과 가장 유사한 암호 작성 방법은?

① 토마스 제퍼슨은 '제퍼슨 바퀴'라는 암호 제조기를 발명하였는데, 이것은 기존의 철자를 다른 철자로 바꾸는 방법을 이용하였다. a, b, c가 F, X, Y로 바뀔 수도 있다.
② 교황의 대사들은 나름대로의 암호를 고안해 냈다. 헤이그의 대사는 MUSEUM을 '바티칸' 대신에 사용했고 비엔나의 대사는 숫자를 이용했다. **7690**은 '나폴레옹'을 가리켰다.
③ 셰익스피어의 작품들은 프란시스 베이컨이 썼을 것이라는 의혹이 제기된 적이 있다. 그렇다면 작품 어디엔 가는 'Bacon'이라는 글자가 존재할 것이다. 파비언은 <사랑의 헛소동> 4막 3장에서, 각 줄의 첫 한두 글자를 따보면, 'B CO AN'이 나온다고 주장하였다.
④ 1943년, 일본의 전쟁 포로 수용소에서 한 사무실로 편지가 왔다. 일본의 검열에도 걸리지 않았지만 FBI에서는 이 편지에 특별한 사연이 있을 것으로 짐작했다. 편지의 각 줄마다 맨 앞에 있는 두 단어만 뽑았더니 다음과 같은 문장이 되었다. "After Surrender fifty percent Americans lost in Philippines in Nippon 30%."

⑤ ISBN 코드는 네 부분이다. '89-89422-42-6'에서 89는 대한민국을 의미하고 89422는 출판사의 고유번호이며 42는 출판사에서 붙인 책의 고유번호이다. 네 번째는 0부터 9까지이며, 10을 쓸 때에는 로마숫자 X를 사용한다. 이 번호는 검사를 위한 수이다. 원래 번호를 쓰고 그 아래에 1부터 10까지를 거꾸로 쓴 다음, 위아래 숫자를 곱해서 이것들을 모두 더하면 11로 나누어지는 숫자가 되어야 한다.

49. 다음 중 ㉡의 이유로 적절한 것은?

① 연감의 통계 수치를 정확하게 보고하기 위하여
② 제3자의 암호 해독 과정을 혼란스럽게 하기 위하여
③ 송신자의 암호문 전문을 다른 코드로 전환하기 위하여
④ 송신자가 정보를 보낼 때 전달 내용의 혼동을 피하기 위하여
⑤ 수신자가 같은 책에서 정확한 열쇠를 뽑아낼 수 있도록 하기 위하여

50. 다음 중 ㉢에 들어갈 숫자에 해당하는 것은?

① 02451 ② 30516 ③ 34236 ④ 36687 ⑤ 72945

- 정답: 47.① 48.① 49.⑤ 50.④

◆ 05-9평 57~60번

[57~60] 다음 글을 읽고 물음에 답하시오.

(가) 인터넷 쇼핑몰에서 물건을 살 때, 다른 사람이 내 컴퓨터와 인터넷 쇼핑몰의 컴퓨터 사이에 오고가는 정보를 읽어서 내가 입력한 신용 카드 정보를 ㉠ 빼내면 어쩌나 하고 걱정하는 사람이 많다. 그러나 공개키 암호화 방식을 이용하면 정보를 주고받는 당사자 이외에는 그 정보를 볼 수 없도록 할 수 있다.

(나) 공개키 암호화 방식에서는 각각의 컴퓨터가 다른 컴퓨터와 절대로 겹치는 법이 없는 한 쌍의 키를 준비한다. 내 컴퓨터가 준비한 키 쌍을 각각 공개키 A와 비밀키 a라고 하자. 공개키 A는 다른 컴퓨터에 알려주는 데에 사용하고 비밀키 a는 내 컴퓨터에만 보관한다. 공개키 A로 암호화된 정보는 오직 비밀키 a가 있어야만 해독되어 원래의 정보로 만들 수 있으며, 공개키 A를 가지고도 해독될 수 없다. 따라서 비밀키 a만 내 컴퓨터 밖으로 빠져나가지 않게 하면 공개키 A는 다른 컴퓨터에 알려 주어도 무방하다.

(다) 이제 인터넷 서점 '책마을'에 접속하여 책을 구매하는 경우를 생각해 보자. 책마을 컴퓨터가 공개키 B와 비밀키 b를 가지고 있다고 하면, 내 컴퓨터가 책마을 컴퓨터에 접속하자마자 두 컴퓨터는 자동적으로 자신들의 공개키를 교환한다. 즉 내 컴퓨터는 B를, 책마을 컴퓨터는 A를 알게 되는 것이다. 이제 내가 책을 주문하기 위해서 신용 카드 정보를 내 컴퓨터에 입력하면 내 컴퓨터는 이것을 책마을 컴퓨터의 공개키 B로 암호화하여 전송한다. 책마을 컴퓨터는 암호화된 정보를 자신의 비밀키 b로 해독하여 원래의 신용 카드 정보를 얻는다. 공개키 B로 암호화하여 보내진 정보는 비밀키 b를 갖고 있는 책마을 컴퓨터만 해독할 수 있으므로 다른 사람이 내 신용 카드 정보를 해독하기는 불가능하다.

(라) 내 컴퓨터의 공개키 A는 다른 컴퓨터에서도 알 수 있으므로 다른 사람이 나인 척하고 자기 컴퓨터에서 공개키 A를 알려주고 책을 주문한다면 곤란한 문제가 발생할 수 있다. 이러한 문제를 방지하기 위해서는 책마을 컴퓨터가 받고 있는 정보의 송신자가 내 컴퓨터라는 것을 확인해야 한다. 이를 위해서 책마을 컴퓨터는 내 컴퓨터에 '책마을만세'와 같은 임의의 단어를 보내면서 이 단어를 내 컴퓨터의 비밀키 a로 암호화한 후, 원래 단어와 암호화된 단어를 함께 보내달라고 요구한다. 공개키 암호화 방식에서는 비밀키 a로 암호화된 정보가 공개키 A로만 해독이 가능하다. 따라서 ㉡ 내 컴퓨터는 원래의 단어와 암호화된 단어를 함께 전송하고, 이 두 정보를 전송 받은 책마을 컴퓨터는 암호화된 단어를 공개키 A로 해독한 후에 전송 받은 원래 단어와 일치하는지 확인한다. 만약 이들이 일치한다면 공개키 A를 가진 컴퓨터(내 컴퓨터)가 보낸 정보임에 틀림없다는 것을 알 수 있다.

(마) 어떤 사람은 자기 컴퓨터가 가르쳐 준 공개키 A에서 비밀키 a를 알아내면 어쩌나 하고 걱정할지 모른다. 그러나 이러한 일은 기술적으로만 본다면 거의 불가능하다. 비밀키 a에서는 간단한 계산만으로 공개키 A를 얻을 수 있다. 그러나 공개키 A에서 비밀키 a를 구하기 위해서는 현재 가장 속도가 빠른 슈퍼컴퓨터를 동원하더라도 수십 년 동안 계산해야 할 정도로 엄청난 시간을 필요로 한다. 따라서 공개키 암호화 방식은 일반적으로 사람들이 안심하고 사용해도 좋다고 할 수 있다.

57. 각 단락의 중심 화제로 적절하지 않은 것은? [1점]

① (가): 공개키 암호화 방식의 효용성
② (나): 공개키와 비밀키를 생성하는 방법
③ (다): 공개키 암호화 방식의 동작 원리
④ (라): 송신자 컴퓨터를 확인하는 원리
⑤ (마): 공개키 암호화 방식의 안전성

58. 위 글에 나타난 '공개키 암호화 방식'에 대한 설명으로 가장 적절한 것은?

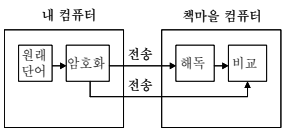
① 정보를 주고받는 컴퓨터끼리는 공통의 비밀키를 사용한다.
② 공개키로 암호화하여 보내는 정보는 비밀키로 해독될 수 없다.
③ 컴퓨터의 속도가 빨라지면 공개키 암호화의 안전성은 높아진다.
④ 정보를 주고받는 컴퓨터끼리는 상대방 컴퓨터의 비밀키를 모르고 있다.
⑤ 공개키로 암호화된 정보는 암호화에 사용된 공개키를 알면 해독될 수 있다.

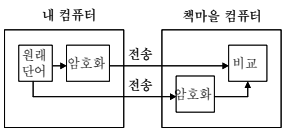
59. 국어 사전에서 ㉠의 의미를 바르게 찾은 것은? [1점]

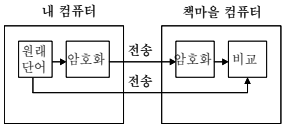
> **빼-내다**[빼 : --] [-내어(-내), -내니] 동 (…에서 …을) (1) 박혀 있거나 끼워져 있는 것을 뽑다. (2) 여럿 가운데에서 필요한 것 혹은 불필요한 것만을 골라내다. (3) 남의 물건 따위를 돌려내다. (4) 남을 꾀어서 나오게 하다. (5) 얽매인 사람을 자유롭게 해 주다.

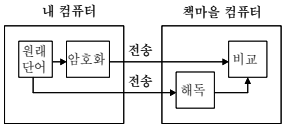
① (1) ② (2) ③ (3) ④ (4) ⑤ (5)

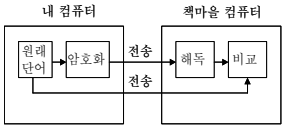
60. ㉡의 내용을 그림으로 올바르게 표현한 것은? [3점]

①


②


③


④


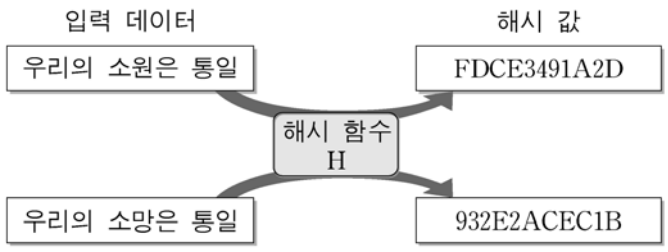
⑤


- 정답: 57.② 58.④ 59.③ 60.⑤

◆ 16-9평 A형 16~18번

[16 ~ 18] 다음 글을 읽고 물음에 답하시오.

온라인을 통한 통신, 금융, 상거래 등은 우리에게 편리함을 주지만 보안상의 문제도 안고 있는데, 이런 문제를 해결하기 위하여 암호 기술이 동원된다. 예를 들어 전자 화폐의 일종인 비트코인은 해시 함수를 이용하여 화폐 거래의 안전성을 유지한다. 해시 함수란 입력 데이터 x에 대응하는 하나의 결과 값을 일정한 길이의 문자열로 표시하는 수학적 함수이다. 그리고 입력 데이터 x에 대하여 해시 함수 H를 적용한 수식을 H(x)=k라 할 때, k를 해시 값이라 한다. 이때 해시 값은 입력 데이터의 내용에 미세한 변화만 있어도 크게 달라진다. 현재 여러 해시 함수가 이용되고 있는데, 해시 값을 표시하는 문자열의 길이는 각 해시 함수마다 다를 수 있지만 특정 해시 함수에서의 그 길이는 고정되어 있다.



[해시 함수의 입 · 출력 동작의 예]

이러한 특성을 갖고 있기 때문에 해시 함수는 데이터의 내용이 변경되었는지 여부를 확인하는 데 이용된다. 가령, 상호 간에 동일한 해시 함수를 사용한다고 할 때, 전자 문서와 그 문서의 해시 값을 함께 전송하면 상대방은 수신한 전자 문서에 동일한 해시 함수를 적용하여 결과 값을 얻은 뒤 전송받은 해시 값과 비교함으로써 문서가 변경되었는지 확인할 수 있다.

그런데 해시 함수가 ㉠ 일방향성과 ㉡ 충돌회피성을 만족시키면 암호 기술로도 활용된다. 일방향성이란 주어진 해시 값에 대응하는 입력 데이터의 복원이 불가능하다는 것을 말한다. 특정 해시 값 k가 주어졌을 때 H(x)=k를 만족시키는 x를 계산하는 것이 매우 어렵다는 것이다. 그리고 충돌회피성이란 특정 해시 값을 갖는 서로 다른 데이터를 찾아내는 것이 현실적으로 불가능하다는 것을 의미한다. 서로 다른 데이터 x, y에 대해서 H(x)와 H(y)가 각각 도출한 값이 동일하면 이것을 충돌이라 하고, 이때의 x와 y를 충돌쌍이라 한다. 충돌회피성은 이러한 충돌쌍을 찾는 것이 현재 사용할 수 있는 모든 컴퓨터의 계산 능력을 동원하더라도 그것을 완료하기가 사실상 불가능하다는 것이다.

[가] 해시 함수는 온라인 경매에도 이용될 수 있다. 예를 들어 ○○ 온라인 경매 사이트에서 일방향성과 충돌회피성을 만족시키는 해시 함수 G가 모든 경매 참여자와 운영자에게 공개되어 있다고 하자. 이때 각 입찰 참여자는 자신의 입찰가를 감추기 위해 논스*의 해시 값과, 입찰가에 논스를 더한 것의 해시 값을 함께 게시판에 게시한다. 해시 값 게시 기한이 지난 후 각 참여자는 본인의 입찰가와 논스를 운영자에게 전송하고 운영자는 최고 입찰가를 제출한 사람을 낙찰자로 선정한다. 이로써 온라인 경매 진행 시 발생할 수 있는 다양한 보안상의 문제를 해결할 수 있다.

* 논스: 입찰가를 추측할 수 없게 하기 위해 입찰가에 더해지는 임의의 숫자.

16. 윗글의 '해시 함수'에 대한 이해로 적절하지 않은 것은?

① 전자 화폐를 사용한 거래의 안전성을 위해 해시 함수가 이용될 수 있다.
② 특정한 해시 함수는 하나의 입력 데이터로부터 두 개의 서로 다른 해시 값을 도출하지 않는다.
③ 입력 데이터 x를 서로 다른 해시 함수 H와 G에 적용한 H(x)와 G(x)가 도출한 해시 값은 언제나 동일하다.
④ 입력 데이터 x, y에 대해 특정한 해시 함수 H를 적용한 H(x)와 H(y)가 도출한 해시 값의 문자열의 길이는 언제나 동일하다.
⑤ 발신자가 자신과 특정 해시 함수를 공유하는 수신자에게 어떤 전자 문서와 그 문서의 해시 값을 전송하면 수신자는 그 문서의 변경 여부를 확인할 수 있다.

17. 윗글의 ㉠과 ㉡에 대하여 추론한 내용으로 가장 적절한 것은?

① ㉠을 지닌 특정 해시 함수를 전자 문서 x, y에 각각 적용하여 도출한 해시 값으로부터 x, y를 복원할 수 없다.
② 입력 데이터 x, y에 특정 해시 함수를 적용하여 도출한 문자열의 길이가 같은 것은 해시 함수의 ㉠ 때문이다.
③ ㉡을 지닌 특정 해시 함수를 전자 문서 x, y에 각각 적용하여 도출한 해시 값의 문자열의 길이는 서로 다르다.
④ 입력 데이터 x, y에 특정 해시 함수를 적용하여 도출한 해시 값이 같은 것은 해시 함수의 ㉡ 때문이다.
⑤ 입력 데이터 x, y에 대해 ㉠과 ㉡을 지닌 서로 다른 해시 함수를 적용하였을 때 도출한 결과 값이 같으면 이를 충돌이라고 한다.

18. [가]에 따라 <보기>의 사례를 이해한 내용으로 가장 적절한 것은? [3점]

<보 기>

온라인 미술품 경매 사이트에 회화 작품 △△이 출품되어 A와 B만이 경매에 참여하였다. A, B의 입찰가와 해시 값은 다음과 같다. 단, 입찰 참여자는 논스를 임의로 선택한다.

입찰 참여자	입찰가	논스의 해시 값	'입찰가+논스'의 해시 값
A	a	r	m
B	b	s	n

① A는 a, r, m 모두를 게시 기한 내에 운영자에게 전송해야 한다.
② 운영자는 해시 값을 게시하는 기한이 마감되기 전에 최고가 입찰자를 알 수 없다.
③ m과 n이 같으면 r과 s가 다르더라도 A와 B의 입찰가가 같다는 것을 의미한다.
④ A와 B 가운데 누가 높은 가격으로 입찰하였는지는 r과 s를 비교하여 정할 수 있다.
⑤ B가 게시판의 m과 r을 통해 A의 입찰가 a를 알아낼 수도 있으므로 게시판은 비공개로 운영되어야 한다.

- 정답: 16.③ 17.① 18.②

◆ 22년 3월 고3 10~13번

[10 ~ 13] 다음 글을 읽고 물음에 답하시오.

문자 입력 창에 한 글자만을 입력했는데 완성된 문구가 ⓐ 제시되는 자동 완성을 경험해 보았을 것이다. '코'라는 문자를 입력했다면 '코피', '코로나' 등이 후보로 제시되어 휴대 전화와 같이 문자 입력이 불편한 경우 문자 입력을 편리하게 할 수 있다. 이는 사용했던 단어들 중에서 입력되는 문자와 첫 글자부터 일치하는 것을 찾고 그중 사용 빈도가 높은 단어들을 후보로 제시하는 것이라고 할 수 있다. 한편 워드 프로세서에서 단어 찾기와 같은 검색은 저장되어 있는 문자열을 대상으로 검색어가 ⓑ 포함된 문자열을 찾는 것이다. 검색은 자동 완성과 달리 대상 문자열의 어느 위치에서도 검색어를 찾을 수 있어야 하며 사용 빈도를 고려하지 않아도 된다.

검색이 가능하기 위해서는 검색어를 저장되어 있는 문자열의 부분 문자열과 비교하는 알고리즘이 필요하다. 예를 들어 '우리글'이라는 검색어를 '한글:␣우리나라에서␣창제된␣우리글'이라는 띄어쓰기(␣)가 포함된 18글자의 대상 문자열에서 검색한다고 ⓒ 가정해 보자. ㉠ 가장 간단히 떠올릴 수 있는 방법은 '우리글'이 3글자이므로 대상 문자열을 3글자씩 잘라 1글자씩 비교하는 것이다. '한글:', '글:␣', ':␣우' 등과 같이 16개의 비교 대상을 만들고 이를 검색어와 각각 비교하여 모두 같은지 확인한다. 하나의 비교 대상을 확인하기 위해서는 3글자를 각각 비교해야 하므로 총 16×3번 비교를 하게 될 것이다. 검색어 길이에 비해 대상 문자열이 짧거나 같은 경우는 없으므로 이 방법은 검색어와 비교해야 하는 대상 문자열의 길이가 길어지거나 개수가 많아지면 비교 횟수가 늘어나 검색 시간이 늘어난다.

[A] 검색 시간을 줄이기 위한 다른 방법은 없을까? 검색어와 비교 대상을 1글자씩 비교하지 않고 3글자씩 한 번에 비교할 수 있다면 그만큼 비교 횟수가 줄어들게 되어 검색 시간이 줄어들 것이다. 이를 위해 각각의 문자열에 특정 값을 ⓓ 생성하는 함수를 설정할 수 있다. 이런 함수를 해시 함수라고 하고, 어떤 문자열에 대해 해시 함수가 생성한 값을 해시값이라고 한다. 만일 해시 함수가 입력 가능한 문자열에 대해 모두 다른 해시값을 생성한다면 검색어의 해시값과 비교 대상의 해시값을 비교하여 두 문자열이 일치함을 단번에 ⓔ 판단할 수 있다.

앞의 예와 같이 검색어가 3글자이고 18글자의 대상 문자열이 제시된다면 비교 대상은 16개가 만들어진다. 하지만 각 비교 대상에서 문자열 비교는 1번의 해시값 비교로 줄어들기 때문에 전체 비교 횟수는 감소하게 된다. 물론 해시값을 생성하는 해시 함수의 연산이 추가되지만 추가되는 연산 시간이 각 글자 단위의 비교에 필요한 연산 시간보다 짧다면 전체적인 검색 시간은 단축될 수 있다. 이런 이유로 해시 함수는 연산이 간단하면서도 중복되지 않는 해시값을 생성할 수 있어야 한다.

10. 윗글을 통해 알 수 있는 내용으로 적절하지 않은 것은?

① 검색은 저장되어 있는 문자열 전체를 대상으로 검색어가 포함되어 있는지 확인한다.
② 검색은 필요에 따라 각기 다른 문자열에 동일한 해시값을 생성하는 해시 함수를 사용한다.
③ 검색은 저장되어 있는 문자열의 부분 문자열과 검색어를 비교하는 알고리즘을 활용한다.
④ 자동 완성은 사용 빈도를 고려하여 입력되는 문자가 포함된 문자열을 후보로 제시한다.
⑤ 자동 완성은 휴대 전화와 같이 문자 입력이 불편한 경우 문자 입력을 편리하게 할 수 있는 방법이다.

11. [A]를 이해한 내용으로 적절한 것은?

① 검색어의 길이가 짧아진다면 비교 대상의 개수가 줄어들어 해시값 비교 횟수가 증가할 수 있겠군.
② 대상 문자열에 반복되는 글자가 많다면 해시값이 작아져서 해시 함수의 연산 시간이 단축될 수 있겠군.
③ 검색어보다 긴 대상 문자열의 개수가 늘어난다면 비교 대상이 늘어나 해시값 비교 횟수가 증가할 수 있겠군.
④ 대상 문자열이 1개일 경우 검색어의 길이가 짧아진다면 비교 대상의 길이가 줄어들어 해시값 비교 횟수가 감소할 수 있겠군.
⑤ 대상 문자열이 2개일 경우 검색어의 길이가 길어진다면 비교 대상의 개수가 늘어나 해시 함수의 연산 시간이 증가할 수 있겠군.

12. ㉠에 <보기>의 조건을 모두 추가하여 검색한다고 할 때, 이에 대한 설명으로 적절하지 않은 것은? [3점]

< 보 기 >

[조건]
- 검색어에 문장 부호가 포함되지 않는 경우 문장 부호가 있는 부분 문자열은 비교 대상에서 제외한다.
- 검색어에 띄어쓰기가 포함되는 경우 띄어쓰기의 위치가 일치하지 않는 부분 문자열은 비교 대상에서 제외한다.

① '우리␣글'로 검색할 경우 띄어쓰기의 위치가 일치하는 비교 대상 3개가 만들어진다.
② '우리␣글'로 검색할 경우의 비교 횟수보다 '우리글'로 검색할 경우의 비교 횟수가 더 많다.
③ '우리글'로 검색할 경우 비교 대상은 '␣우리', '우리나', '리나라' 등과 같이 3글자로 된 비교 대상들이 만들어진다.
④ '우리글'로 검색할 경우 부분 문자열 '한글:', '글:␣', ':␣우'에는 문장 부호가 포함되어 있기 때문에 비교하지 않는다.
⑤ '우리글'로 검색할 경우 일치하는 문자열을 찾을 수 있지만 '우리␣글'로 검색할 경우는 일치하는 문자열을 찾을 수 없다.

13. ⓐ ~ ⓔ의 사전적 의미로 적절하지 않은 것은?

① ⓐ : 어떠한 의사를 말이나 글로 나타내어 보임.
② ⓑ : 어떤 사물이나 현상 가운데 함께 들어 있거나 함께 넣음.
③ ⓒ : 다른 사람의 말이나 행동, 형편 따위를 잘 알아서 긍정하고 이해함.
④ ⓓ : 사물이 생겨남. 또는 사물이 생겨 이루어지게 함.
⑤ ⓔ : 사물을 인식하여 논리나 기준 등에 따라 판정을 내림.

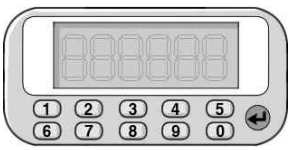
- 정답: 10.② 11.③ 12.① 13.③

◆ 19년 3월 고3 30~33번

[30 ~ 33] 다음 글을 읽고 물음에 답하시오.

인터넷 뱅킹이나 전자 상거래를 할 때 온라인상에서 사용자 인증은 필수적이다. 정당한 사용자인지를 인증받는 흔한 방법은 아이디(ID)와 비밀번호를 입력하는 것으로, 사용자가 특정한 정보를 알고 있는지 확인하는 방식이다. 그러나 이러한 방식은 고정된 정보를 반복적으로 사용하기 때문에 정보가 노출될 수 있다. 이러한 문제점을 보완하기 위해 개발된 인증 기법이 OTP(One-Time Password, 일회용 비밀번호) 기술이다. OTP 기술은 사용자가 금융 거래 인증을 받고자 할 때마다 해당 기관에서 발급한 OTP 발생기를 통해 새로운 비밀번호를 생성하여 인증받는 방식이다.

OTP 기술은 크게 비동기화 방식과 동기화 방식으로 나눌 수 있다. 비동기화 방식은 OTP 발생기와 인증 서버 사이에 동기화된 값이 없는 방식으로, 인증 서버의 질의에 사용자가 응답하는 방식이다. OTP 기술 도입 초기에 사용된 질의 응답 방식은 인증 서버가 임의의 6자리 수, 즉 질읫값을 제시하면 사용자는 그 수를 OTP 발생기에 입력하고, OTP 발생기는 질읫값과 다른 응답값을 생성한다. 사용자는 그 값을 로그인 서버에 입력하고 인증 서버는 입력된 값을 확인한다. 이 방식은 사용자가 OTP 발생기에 질읫값을 직접 입력해 응답값을 구해야 하는 번거로움이 있기 때문에 사용이 불편하다.


<초기 OTP 발생기>

이와 달리 동기화 방식은 OTP 발생기와 인증 서버 사이에 동기화*된 값을 설정하고 이에 따라 비밀번호를 생성하는 방식으로, 이벤트 동기화 방식과 시간 동기화 방식이 있다. 이벤트 동기화 방식은 기촛값과 카운트값을 바탕으로 OTP 발생기는 비밀번호를, 인증 서버는 인증값을 생성하는 방식이다. 기촛값이란 사용자의 신상 정보와 해당 금융 기관의 정보 등이 반영된 고유한 값이며, 카운트값이란 비밀번호를 생성한 횟수이다. 사용자가 인증을 받아야 할 경우 이벤트 동기화 방식의 OTP 발생기는 기촛값과 카운트값을 바탕으로 비밀번호를 생성하게 되며, 생성된 비밀번호를 사용자가 로그인 서버에 입력하면 된다. ㉮ 이때 OTP 발생기는 비밀번호를 생성할 때마다 카운트값을 증가시킨다. 인증 서버 역시 기촛값과 카운트값으로 인증값을 생성하여 로그인 서버로 입력된 OTP 발생기의 비밀번호와 비교하는 것이다. 이때 인증에 성공하면 인증 서버는 카운트값을 증가시켜서 저장해 두었다가 다음번 인증에 반영한다. 그러나 이 방식은 OTP 발생기에서 비밀번호를 생성만 하고 인증하지 않으면 OTP 발생기와 인증 서버 간에 카운트값이 달라지는 문제점이 있다.

시간 동기화 방식은 현재 금융 거래에서 주로 사용되는 방식으로, 기촛값과 인증을 시도한 날짜와 시간을 바탕으로 일정한 시간 간격마다 일방향 함수를 통해 OTP 발생기는 비밀번호를, 인증 서버는 인증값을 생성하는 방식이다. 일방향 함수란 계산하기는 쉽지만 역연산하는 것은 매우 어려운 함수로, 결괏값을 안다고 하더라도 입력값을 구하는 것이 매우 어려운 특성이 있다.

시간 동기화 방식으로 일회용 비밀번호를 생성하는 과정은 다양하지만 다음과 같은 과정을 생각해 볼 수 있다.

[가] 사용자가 인증을 받아야 할 경우 시간 동기화 방식의 OTP 발생기는 발급 시 동기화된 기촛값과 인증 시도 시간을 바탕으로 r를 구하고, r에 대해 일방향 함수 f를 n번 수행하여 X_n을 생성한다. 이렇게 생성된 X_n을 사용자가 로그인 서버에 입력하면, 로그인 서버는 입력된 X_n을 일방향 함수 f로 한 번 더 계산해 X_{n+1}을 구하고 이 값을 인증 서버로 전달하게 된다. 인증 서버 역시 기촛값과 인증 시도 시간을 바탕으로 r를 구하고, r에 대해 일방향 함수 f를 n+1번 수행하여 X_{n+1}을 생성한 후 로그인 서버로부터 전달받은 값과 비교하여 인증을 하게 된다.

시간 동기화 방식의 OTP 발생기에는 인증 서버의 시간과 같은 시간을 가리키는 전자시계가 장착되어 있어 시간 동기화가 가능하다. 하지만 인증 서버와 OTP 발생기 간에 시간 오차가 발생하면 인증에 실패한다. 또한 시간 동기화 방식은 이벤트 동기화 방식에 비해 입력 시간에도 제약을 받는다. 왜냐하면 사용자의 비밀번호 입력 시간이 길어지면 새로운 비밀번호가 생성되기 때문이다.

* 동기화 : 서로 일관성 있게 같은 값을 유지하는 것. 같은 시점에서 특정 작업을 수행하는 것.

30. 윗글에 대한 이해로 가장 적절한 것은?

① 이벤트 동기화 방식은 시간 동기화 방식에 비해 로그인 서버에 비밀번호를 입력해야 하는 시간에 제약을 받지 않는다.
② 비동기화 방식의 OTP 기술에서는 OTP 발생기의 질의에 사용자가 응답값을 인증 서버에 입력해야 인증에 성공한다.
③ 아이디와 비밀번호를 입력하는 방식에서는 고정된 정보를 반복적으로 사용하기 때문에 정보가 노출될 우려가 없다.
④ 시간 동기화 방식에서는 비밀번호 생성 간격을 짧게 할수록 비밀번호가 바뀌는 횟수가 감소할 것이다.
⑤ 질의 응답 방식에서 사용자가 OTP 발생기에 입력한 임의의 6자리 수는 응답값과 일치할 것이다.

31. 윗글을 바탕으로 <보기>를 이해한 내용으로 적절하지 않은 것은?

< 보 기 >

① 시간 동기화 방식에서 인증에 성공하였다면 사용자가 ⓐ에서 ⓑ로 보낸 비밀번호와 ⓑ에서 생성한 인증값은 같을 것이다.
② 시간 동기화 방식에서 ⓐ와 ⓒ 사이에 시간 오차가 발생하면 ⓐ에서 생성한 비밀번호로는 인증에 성공할 수 없을 것이다.
③ 이벤트 동기화 방식에서는 기촛값과 카운트값을 바탕으로 ⓐ는 비밀번호를, ⓒ는 인증값을 생성할 것이다.
④ 이벤트 동기화 방식에서 ⓐ로 비밀번호를 생성하기만 하고 인증하지 않는다면 ⓐ와 ⓒ의 카운트값이 서로 달라질 것이다.
⑤ 이벤트 동기화 방식에서 ⓐ가 생성한 비밀번호로 인증을 받았다면 ⓒ는 카운트값을 증가시켜 다음번 인증에 반영할 것이다.

32. ㉮의 이유로 가장 적절한 것은?

① 비밀번호가 고정되지 않고 새롭게 생성되도록 하기 위해
② 인증 서버의 응답값과 카운트값을 일치시키기 위해
③ 인증에 성공할 때마다 기촛값을 동기화하기 위해
④ 인증에 실패 시 이전 비밀번호를 복원하기 위해
⑤ OTP 발생기의 질읫값이 갱신되도록 하기 위해

33. [가]를 바탕으로 <보기>를 이해한 내용으로 적절하지 않은 것은? [3점]

< 보 기 >

사용자 A와 사용자 B는 모두 각자의 OTP 발생기를 통해 ㉠ 2019년 3월 7일 오전 10 : 00에 인증을 시도하고, ㉡ 오전 10 : 30에 인증을 다시 시도하였다. 그리고 ㉢ 다음날 오전 10 : 30에 다시 인증을 시도하였다.

① ㉠에서 X_n이 노출되더라도 r는 알아내기가 어렵겠군.
② ㉠과 ㉡에서 사용자 A의 r는 서로 다르겠군.
③ ㉡과 ㉢에서 함수f를 n번 수행한 X_n은 같겠군.
④ ㉢에서 사용자 A와 사용자 B의 기촛값은 서로 다르겠군.
⑤ ㉠ ~ ㉢에서 사용자 B의 X_{n+1}들은 서로 다르겠군.

- 정답: 30.① 31.① 32.① 33.③

◆ 24년 9월 고1 21~25번

[21~25] 다음 글을 읽고 물음에 답하시오.

인터넷의 발달로 데이터 저장 및 분석 과정이 인터넷상에서 ⓐ 이루어지고 있으며 그에 따라 개인정보와 같은 민감한 데이터는 암호화되어 인터넷 서버에 저장된다. 그런데 현재 널리 사용되는 공개키 암호화 방식으로 암호화된 데이터는 통계 처리를 위한 연산을 수행하기 위해서 원래 데이터로 복원하는 복호화 과정을 거친 후 연산을 수행하고 그 결과를 다시 암호화해야 한다. 하지만 이 과정에서 비밀키나 민감한 개인정보가 유출되는 일이 생길 수 있다. 그래서 암호화된 데이터를 복호화하지 않고 암호화된 상태로 안전하게 연산을 수행할 수 있는 동형암호가 등장하였다.

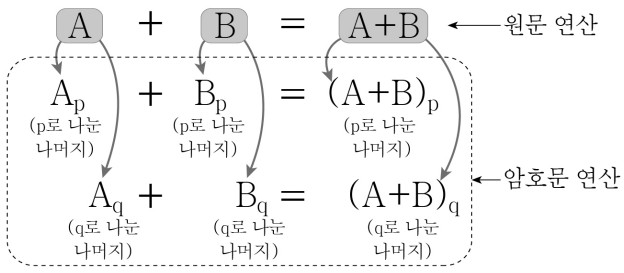
동형암호는 동형성을 기반으로 하는데, 동형성이란 데이터를 암호화한 상태에서 특정 연산을 수행했을 때 나오는 결과가 암호화하지 않은 상태에서 같은 연산을 수행하고 암호화를 한 결과와 같은 것을 ⓑ 말한다. 이때 연산의 횟수에 제한 없이 특정한 한 종류의 연산에만 동형성을 갖는 암호를 부분 동형암호, 연산의 종류와 관계없이 특정 횟수까지만 동형성을 갖는 암호를 제한적 동형암호라고 하며, 횟수에 제한 없이 컴퓨터의 주된 연산인 덧셈, 곱셈에 동형성을 갖는 암호를 완전 동형암호라고 한다.

완전 동형암호는 암호화에 사용하는 원리에 따라 격자 기반, CRT(Chinese Remainder Theorem) 기반 등으로 ⓒ 나뉜다. 그중 ㉠ 격자 기반 완전 동형암호는 수학계에서 답을 찾기 어렵다고 알려진 격자 문제를 응용하여 만들어졌다. 이 방식은 원문 데이터를 비트* 단위로 변환하고 각각의 비트를 개별적으로 암호화한다. 암호키 p와 임의의 정수를 곱한 수를 원문에 더하면 암호문이 만들어지는데, 이 과정에서 무작위로 오륫값을 추가하여 안전성을 높인다. 그래서 암호문의 연산을 반복할수록 오륫값이 커지게 되며, 특히 곱셈 연산을 수행할수록 오륫값이 급격하게 커지기 때문에 일정 횟수 이상 수행하면 원문 복호화가 불가능하다.

따라서 연산을 지속적으로 수행하기 위해서는 오륫값이 한계치에 ⓓ 이른 암호문은 부트스트래핑 과정을 반드시 거쳐야 한다. 일정 횟수의 덧셈과 곱셈 연산을 수행하여 암호문에 오륫값이 누적되면, 다른 암호키로 해당 암호문과 암호키 p를 암호화한다. 그리고 복호화 회로를 통해 기존의 암호키 p에 의한 이전 암호문을 복호화하면 그동안의 연산 과정에서 누적된 오륫값이 제거된 새로운 암호문이 ⓔ 만들어진다. 이때 새로운 암호문이 만들어지면서 오륫값이 추가되지만 그 크기가 기존의 누적된 것보다 작아서 적절하게 부트스트래핑 과정을 수행한다면 지속적인 연산이 가능하다.

이 방식은 원문을 비트 단위로 변환하여 각 비트별로 암호화하기 때문에 원문에 비해 암호문의 값이 10~100배가량 커져서 데이터의 저장 공간이 많이 필요하다. 그리고 개별 비트 단위로 암호문의 연산과 부트스트래핑 과정을 거쳐야 하기 때문에 연산 속도가 느리다.

[A] 그래서 최근에는 효율성을 개선한 ㉡ CRT 기반 완전 동형암호가 등장하였다. 이 방식은 하나의 원문을 특정한 정수인 암호키로 나눈 나머지 값을 암호문으로 이용하고, 이 나머지 값에서 원문을 복호화하는 방법이다. 이때 암호키의 개수는 임의로 설정할 수 있으며 각각의 원문마다 암호키의 개수만큼 암호문이 만들어진다. 암호키가 두 개일 때 정수로 된 원문 A와 B를 덧셈 연산한 결과가 동형성을 갖는 원리를 간단히 알아보자. 우선 서로소*인 임의의 정수 p와 q를 암호키로 정하고 정수로 된 원문 A와 B를 각각의 암호키로 나눈 나머지 값을 구하면 A_p, A_q와 B_p, B_q가 되는데 이 나머지 값이 원문 A와 B의 암호문이 된다. 그리고 <그림>처럼 각 원문을 동일한 암호키로 나눈 나머지 값인 A_p와 B_p, A_q와 B_q끼리 서로 덧셈 연산을 수행한다. 만약 연산 수행의 결괏값이 암호키와 같거나 암호키보다 크면 한 번 더 암호키로 나누어 나머지 값을 구한다. 그러면 연산 수행의 결괏값인 A_p+B_p, A_q+B_q가 원문 A와 B를 직접 덧셈 연산한 결괏값을 암호키 p와 q로 나눈 나머지 값인 $(A+B)_p$, $(A+B)_q$와 같다. 그리고 원문을 각 암호키로 나누었을 때의 나머지 값과 각 암호키를 알면 원문을 복호화할 수 있다.



<그림>

이 방식 또한 안전성을 위해서 암호키의 개수를 늘려 계산이 복잡하고 무작위로 오륫값을 추가하기 때문에 부트스트래핑 과정이 필요하다. 하지만 데이터를 정수 단위로 암호화하기 때문에 비트 단위로 암호화하는 격자 기반의 방식보다 더 많은 데이터를 저장할 수 있다. 또한 CRT 방식은 원문보다 작은 나머지 값으로 연산을 수행하기 때문에 격자 기반의 방식에 비해 연산 값이 상대적으로 작아 연산 속도가 빠르고, 격자 기반의 방식과 달리 병렬적으로 연산을 수행할 수 있다.

* 비트: 정보량의 최소 기본 단위. 1비트는 이진수 체계(0, 1)의 한 자리.
* 서로소: 여러 개의 수 사이에 1 이외의 공약수가 없음을 이르는 말.

21. 윗글의 내용과 일치하지 않는 것은?

① 제한적 동형암호는 컴퓨터의 특정한 한 종류의 연산에만 동형성을 갖는 암호이다.
② 격자 기반 완전 동형암호는 수학적으로 답을 찾기 어려운 문제를 응용하여 만들어졌다.
③ 공개키 방식으로 암호화된 데이터를 연산하기 위해서는 원래의 데이터로 복호화해야 한다.
④ CRT 기반 완전 동형암호는 원문을 특정한 정수로 나눈 나머지 값을 암호문으로 사용한다.
⑤ 격자 기반 완전 동형암호는 암호키와 임의의 정수를 곱한 수를 원문에 더해서 암호문을 만든다.

22. 부트스트래핑에 대해 이해한 내용으로 적절하지 않은 것은?

① 부트스트래핑은 동일한 암호문을 연산할 때 덧셈 연산보다 곱셈 연산을 많이 수행할수록 더 빨리 시작된다.
② 부트스트래핑은 암호문의 연산 과정에서 오륫값이 한계치에 이르렀을 때 진행된다.
③ 부트스트래핑에 사용되는 암호키는 이전 암호화에 사용된 암호키와 다르다.
④ 부트스트래핑의 과정을 거치면 이전 암호화된 암호문이 복호화된다.
⑤ 부트스트래핑의 결과로 생성된 새로운 암호문에는 오륫값이 없다.

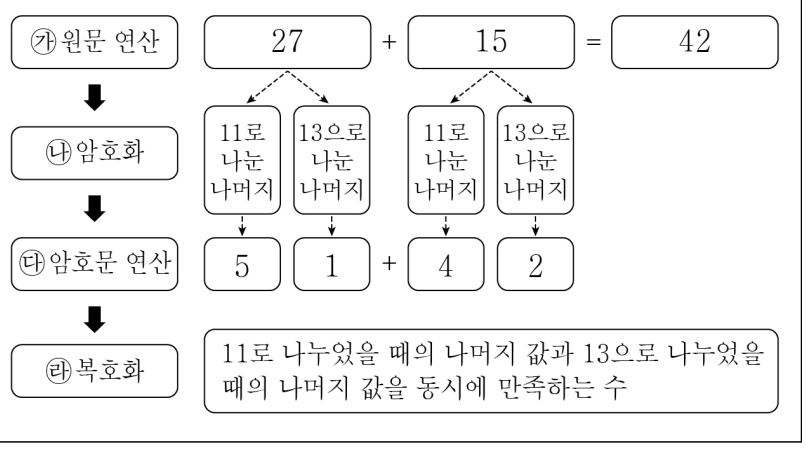
23. ㉠과 ㉡을 비교하여 이해한 내용으로 적절하지 <u>않은</u> 것은?

① ㉠은 ㉡과 달리 비트 단위로 암호문 연산을 수행한다.
② ㉠은 ㉡과 달리 원문을 암호화했을 때 암호문의 값이 원문보다 커진다.
③ ㉡은 ㉠과 달리 암호문에 오륫값을 추가하여 안전성을 높인다.
④ ㉡은 ㉠과 달리 데이터를 병렬적으로 연산하는 것이 가능하다.
⑤ ㉠과 ㉡은 모두 암호문을 연산하는 횟수에 제한이 없다.

24. [A]를 바탕으로 <보기>를 이해한 내용으로 가장 적절한 것은? [3점]

<보 기>

다음은 CRT 기반 완전 동형암호를 활용하여 연산하는 과정이다.



① ㉮에서 원문 연산의 결괏값을 암호키로 암호화하면 5, 4가 된다.
② ㉯에서 각 원문을 암호화한 암호키의 개수는 총 4개이다.
③ ㉰에서 만들어진 암호문을 연산한 결괏값은 암호키로 다시 나눌 필요가 없다.
④ ㉱에서 암호키를 알면 나머지 값을 몰라도 원문 27과 15를 복호화할 수 있다.
⑤ ㉮~㉰의 과정을 통해 만들어진 연산 결괏값은 암호문과 달리 정수이다.

25. 문맥상 ⓐ~ⓔ와 바꾸어 쓰기에 가장 적절한 것은?

① ⓐ: 달성(達成)되고
② ⓑ: 제시(提示)한다
③ ⓒ: 분리(分離)된다
④ ⓓ: 도달(到達)한
⑤ ⓔ: 결성(結成)된다

- 정답: 21.① 22.⑤ 23.③ 24.③ 25.④

◆ 19년 7월 고3 33~37번

[33~37] 다음 글을 읽고 물음에 답하시오.

온라인 전자 상거래나 공인 인증이 일상화되면서 보안을 위해 메시지를 암호화하여 주고받는 암호통신의 중요성이 강조되고 있다. 암호통신에서 가장 핵심적인 문제 중 하나는 메시지를 암호화하거나 이를 다시 원래의 메시지로 복호화하는 데 필요한 키를 암호통신의 대상자인 송·수신자가 어떻게 안전하게 주고받느냐에 대한 것이다. 이러한 암호통신은 암호화나 복호화에 필요한 키를 관리하는 방식에 따라 크게 ㉠ 대칭키 방식과 ㉡ 공개키 방식으로 구분된다.

대칭키 방식은 메시지를 암호화하거나 복호화할 때 동일한 키를 사용한다. 이러한 이유로 송신자와 수신자만 아는 비밀키를 미리 분배하고 사용하는 과정에서 키 정보가 유출될 가능성이 높아 암호통신을 시도할 때마다 상대에 따라 새로운 비밀키를 사용해야 한다. 이에 반해 공개키 방식은 암호화 키와 복호화 키가 서로 다른 방식이다. 수신자가 미리 생성하여 공개한 공개키(public key)로 송신자가 메시지를 암호화하여 전송하면 수신자는 공개키에 대응하여 생성한, 자신만 알고 있는 비밀키(private key)를 이용하여 복호화한다. 공개키 방식은 별도의 비밀키 분배 과정이 필요 없고 통신 상대에 따라 비밀키를 바꿀 필요도 없어 대칭키 방식에 비해 보안에 유리하다.

대표적인 공개키 방식인 RSA 알고리즘은 큰 소수의 곱과 추가 연산을 통해 만들어진 정수의 소인수 분해가 매우 어렵다는 점에 기반하여 한 쌍의 공개키와 비밀키를 생성한다. 키를 만드는 연산 과정이 복잡하여 대칭키 방식에 비해 암호화나 복호화 속도가 상대적으로 느리지만 암호화된 문서가 유출되어도 현재의 컴퓨터 성능으로는 비밀키를 유추하는 데 비현실적으로 오랜 시간이 걸리기 때문에 비밀키를 바꿀 필요가 없다. 하지만 컴퓨터 연산 속도가 급격하게 발전하게 되면 복잡한 연산 과정을 기반으로 한 공개키 방식의 암호 체계가 위협받을 가능성이 높아질 수 있다.

그래서 최근 수학적 복잡성에 의존하지 않으면서도 도청으로부터 비밀키를 안전하게 나누어 가질 수 있는 ㉢ 양자암호통신 기술이 주목받고 있다. 양자암호통신에서는 매번 새롭게 만들어지는 비밀키를 안전하게 나누어 갖기 위해 양자의 종류 중 하나인 광자의 물리적 특성을 이용한다. 원자나 분자 단위 이하의 미시 세계를 다루는 양자 역학에서 광자는 더 이상 나눌 수 없는 최소 단위이기 때문에 광자 하나하나에 정보를 실어 보내는 양자암호통신에서 단일광자에 실린 정보의 일부만을 가로채는 것은 불가능하다. 또한 도청자가 단일광자 자체를 가로챈다 하더라도 수신자에게 가로챈 광자와 동일한 상태의 광자를 보내야만 도청 사실을 숨길 수 있는데 여러 상태를 동시에 지니는 '중첩'이라는 양자의 특성 때문에 단일광자의 원래 상태를 정확히 측정해 보낼 수 없다. 이러한 이유들로 인해 양자암호통신은 도청으로부터 안전한 신호 전달이 가능하다.

양자암호통신의 대표적인 키 분배 기술로는 단일광자의 편광 상태에 정보를 실을 수 있는 BB84 프로토콜*을 들 수 있다. 자연 상태의 빛은 진행하는 방향과 수직인 모든 방향으로 진동하는 특성이 있는데, 진동 방향에 따라 빛을 선택적으로 통과시킬 수 있는 필터를 이용하면 특정한 방향으로 진동하는 빛을 만들 수 있다. 이러한 빛을 '편광'이라고 하며, 편광을 만들 때 이용하는 필터를 '편광필터'라고 한다. 그런데 편광된 광자 또한 여러 방향으로 진동하는 '중첩' 특성을 지니고 있다. 즉 편

광필터를 통과한 수직(↕)이나 수평(↔) 편광의 경우 대각(↗)·역대각(↖) 편광 특성도 지니고 있으며, 마찬가지로 편광필터를 통과한 대각이나 역대각 편광 또한 수직·수평 편광 특성을 동시에 지니고 있다. 따라서 수직이나 수평 편광을 ✚편광필터를 이용하여 측정하면 수직이나 수평 편광으로 100% 측정되지만, 수직이나 수평 편광을 ✖편광필터를 이용하여 측정하면 대각 혹은 역대각 편광으로 잘못 측정된다.

[A] 이러한 편광의 중첩 특성이 BB84 프로토콜에서 어떻게 이용되는지 알아보자.

(a) 송신자의 비트 정보	0	1	1	0	1	0
(b) 송신자의 편광필터	✚	✚	✖	✚	✖	✖
(c) 송신자의 편광 신호	↔	↕	↘	↔	↘	↗
(d) 수신자의 편광필터	✚	✚	✖	✖	✚	✖
(e) 수신자의 측정 신호	↔	×	↘	↗	↕	↗
(f) 비밀키 공유	0		1			0

※'×'는 누락된 광자.

BB84 프로토콜은 먼저 위 <표>의 (a)처럼 송신자가 무작위로 비트 정보를 생성하는 것으로 시작한다. 이때 BB84 프로토콜은 수직 편광과 역대각 편광은 '1'이라는 비트 정보로, 수평 편광과 대각 편광은 '0'이라는 비트 정보로 표시하기로 약속되어 있어 (b)처럼 송신자가 ✚편광필터와 ✖편광필터를 무작위로 선정하면 (c)와 같은 편광 신호들이 생성된다. 수신자는 (c)에서 생성된 편광 신호들이 어떤 편광인지 전혀 모르는 상태에서 (d)처럼 스스로 무작위로 편광필터를 선택하여 (e)와 같이 편광된 광자를 측정한다. 이때 전송 과정에서 잡음 등으로 인해 누락된 광자가 발생할 수 있으며, 누락된 광자는 측정에서 제외된다. 이후 송·수신자는 공개 채널에서 자신들이 어떤 편광필터를 어떤 순서로 사용했는지 서로 공유하면 (f)와 같이 동일한 편광필터를 사용한 '010'이라는 비트 정보만 걸러낼 수 있어 비밀키로 사용하는 측정값을 안전하게 공유할 수 있다.

* 프로토콜 : 통신 규약.

33. 다음은 윗글을 읽은 학생의 독서 기록 중 일부이다. 윗글을 참고할 때, '점검 결과'로 적절하지 않은 것은?

○ 읽기 계획 : 1문단을 훑어보면서 뒷부분을 예측하고 질문 만들기를 한 후 글을 읽고 점검하기

예측 및 질문 내용	점검 결과
○ 암호통신을 이용하여 온라인 전자 상거래가 이루어지는 과정을 보여 줄 것이다.	예측과 다름 ····· ①
○ 암호통신 방식에 따른 장단점을 비교하며 설명할 것이다.	예측과 같음 ····· ②
○ 암호화 키를 만드는 방법은 복호화 키를 만드는 방법과 어떠한 차이가 있을까?	질문의 답이 제시됨 ····· ③
○ 암호통신 방식에 따라 안전성을 확보하기 위한 방법은 어떻게 다를까?	질문의 답이 제시됨 ····· ④
○ 각각의 암호통신 방식이 실생활에 적용된 사례로는 어떤 것이 있을까?	질문의 답이 언급되지 않음 ····· ⑤

34. 윗글에 대한 이해로 적절하지 않은 것은?

① 공개키 방식에서 공개키와 비밀키를 생성하는 주체는 동일하겠군.
② 컴퓨터의 연산 능력이 발전하더라도 양자암호통신은 비밀키를 안전하게 나누어 가질 수 있겠군.
③ 양자암호통신에서는 도청자가 단일광자에 담긴 정보를 도청할 경우 수신자에게 도청 사실을 숨길 수 없겠군.
④ RSA 알고리즘에서 암호화된 문서가 전송 과정 중 유출되어도 수신자는 비밀키를 다시 생성할 필요가 없겠군.
⑤ RSA 알고리즘이 대칭키 방식에 비해 암·복호화 속도가 느린 이유는 서로 다른 암·복호화 키를 주고받기 때문이겠군.

35. ㉠~㉢을 비교한 내용으로 가장 적절한 것은?

① ㉠은 ㉡이나 ㉢에 비해 비밀키가 유출될 가능성이 낮다.
② ㉢은 ㉠이나 ㉡에 비해 수학적 복잡성에 더 많이 의존한다.
③ ㉠과 ㉡은 ㉢과 달리 비밀키를 나누어 갖는 과정이 필요하다.
④ ㉠과 ㉢은 ㉡과 달리 암호화를 위해 송신자가 비밀키를 알아야 한다.
⑤ ㉠, ㉡, ㉢은 모두 암호통신 상대의 수만큼 비밀키가 필요하다.

36. BB84 프로토콜에 대한 이해로 가장 적절한 것은?

① BB84 프로토콜은 안전한 비밀키를 사용하여 암·복호화를 하는 과정에 대한 통신 규약이다.
② BB84 프로토콜에 사용되는 수평 편광을 ✖편광필터로 측정하면 수평 편광으로 측정되지 않는다.
③ BB84 프로토콜 실행 과정에서 편광된 광자가 다시 편광필터를 통과하면 양자의 중첩 특성이 사라진다.
④ 광자는 더 이상 나눌 수 없기 때문에 BB84 프로토콜이 진행되는 동안 단일광자 자체를 가로챌 수 없다.
⑤ BB84 프로토콜에서 수직 편광은 대각 편광의 특성도 동시에 지니고 있어 '0'이라는 비트 정보로 표현한다.

37. BB84 프로토콜을 이용하여 송신자와 수신자가 <보기>와 같이 정보를 주고받았다. [A]를 참고했을 때 <보기>의 과정을 통해 생성되는 비밀키로 적절한 것은? [3점]

<보 기>

○ 송신자의 비트 정보 생성 및 편광된 광자 전송

비트 정보	0	1	0	0	1	1	1	0	1	0
편광필터 정보	0	1	1	0	1	0	1	1	1	0
편광 신호	↔	↘	↗	↔	↘	↕	↘	↗	↘	↔

○ 수신자의 광자 측정

편광필터 정보	1	1	0	1	1	0	0	1	1	1
측정한 신호	↘	↘	↕	↘	×	↕	↔	↗	↘	↗

* ✚편광필터 : 0, ✖편광필터 : 1, 누락된 광자 : ×

① 1011 ② 1100 ③ 1101 ④ 11011 ⑤ 11101

- 정답: 33.③ 34.⑤ 35.④ 36.② 37.③