흘겨보기 7쪽 예정

바라보기

‘난 여기있는데~ 나를 몰라보나요~~’[1], 휴대전화에서 울린 알람을 듣고 잠에서 깨어났다. “으음~ 형석이한테 전화왔네. 무슨 일이지. 이따 톡으로 오겠지.” 아침잠에서 헤어나오지 못한 채, 밥과 김치를 꺼내어, 오늘 나온 웹툰을 보며 먹는 둥 마는 둥 먹는다. 혼자라도 외롭지 않다. 화장실에 앉아, 오늘 뉴스 기사 헤드라인을 읽는다.

‘XX사건 범인 인터넷 아이디 추적 끝에 검거 성공.’

‘블록체인 시스템 차세대 금융 시스템으로 각광’

‘국민청원 20만 명 돌파, 대통령 ‘문제가 있음을 인지, 해결하도록 노력’

‘뭐야 이런 청원도 있었어? 억울하겠다. 나도 동의해야지.’ 뉴스를 보다보니 늦어 버렸다. ‘아 나 9시 수업인데.’ 아무리 늦었어도 노래는 들으면서 가야 한다. 이어폰을 귀에 꽂은 채, 수업에 헐레벌떡 뛰어간다. 뛰어왔더니 피곤하다. ‘수업은 폰 녹음기가 대신 들어줄 테니 1교시에는 눈을 좀 붙여야겠다.’ 폰에는 녹음 파일 4개가 밀려 있다. ‘아이씨, 언제 다 들어.’ 오늘은 아침 수업뿐이다. 점심은 학교 앞 맥도날드[2]에서 여자친구랑 먹기로 했다. ‘XX아, 너는 뭐먹을래?’.‘나는 ~~’. 키오스크로 주문을 하고 점원에게 완성된 햄버거를 받아 온다. 잠깐만 기다려. 사진 좀 찍게. 오늘도 역시 점심 먹는 사진을 찍어 인스타그램에 올린다.

---

1) 김창락 외 2명, Goodbye, 2018
2) 우리 학교 근처에는 맥도날드가 없다.

해시태그

#フォロ一バック#いいねした人全員ファロ一する#いいね返し#いいね返します#좋아요테러#좋아요반사#팔로우#맞팔환영#맞팔#선팔하면맞팔#선팔맞팔#좋아요테러#좋반#좋아요#좋아요반사#팔로우#맞팔환영#맞팔해요#선팔하면맞팔#선팔#follow4like#fff#followforfollowback#likeforlikes#like4follow#l4f#f4f#l4l#ootd#다렉#韩国人#留学生#美女#맞팔환영#일상#선팔하면맞팔#가로수길#대학생#훈녀#훈남#친스타그램#데일리#팔로미#오늘의훈남#셀기꾼#makeup#뷰티#뷰티스타그램#메이크업#팔로우미#협찬#좋반테러#댓글#인스타데일리#다이렉트#첫줄#아웃핏#ootd

금새 팔로워가 23명 늘었다. 나도 어느새 팔로워가 1500명을 넘었다. 밥은 대충 햄버거로 때웠어도, 카페는 예쁜데에 가서 인스타그램에 올려야 한다. 여자친구가 요즘 유명하다는 카페에 가기로 하고 맥도날드를 나선다. 생각해보니 요즘 카페는 음료보다는 인테리어가 중요한거 같다.

오후 스토리

흘겨보기

‘난 여기있는데~ 나를 몰라보나요~~’[3], 어우... 지금이 몇시야... 잠에서 일어나 폰을 확인한다. 지금은 8시. ‘에이, 어제 유튜브보다가 4시간 밖에 못잤네...’. 옷을 주섬주섬 챙겨입고, 웹툰을 보면서 아침을 먹는다. 생각해보니 자취한 이후로 아침을 끼니답게 챙겨먹은 적이 없는 거 같다. 화장실에 앉아 뉴스 기사 헤드라인을 읽는다.

‘XX사건, 인터넷으로 피해자 신상 해킹한 것으로 밝혀져’

‘비트코인 투기 과열 현상, 파산 신청 잇따라’

‘국민청원 당사자 자살, 유서에 억울함 호소’

요즘 사회가 왜 이렇게 시끄러워...

‘흐음... 이런 기사는 댓글 보는 맛이 있지.’

‘에휴, 도둑이 제 발 저린다고 찔려서 뒤진거지ㅋㅋㅋㅋㅋ’

‘중립기어 넣자니까. 선동으로 사람 한 명 죽이네 ㄷㄷ’

나도 한 마디를 덧붙인다. ‘ㄹㅇㅋㅋ, 역시 떼법의 나라’ 아 늦었다. 오늘도 화장실에 20분이나 앉아있었다. 1교시 수업이 있는데 시작까지 10분밖에 안 남았다. 하지만, 아무리 늦어도 노래는 들으면서 가야한다. 이어폰으로 노래를 듣느라 하마터면 사고날 뻔 했다. 수

3) 김창락 외 2명, Goodbye, 2018

업에 헐레벌떡 들어가 수업을 듣기 시작한다. 그래도 역시 수업은 녹음기의 몫이다. 올해 수업을 직접 들어본 기억이 없는 것 같다. 오늘은 여자친구랑 밥을 먹기로 했다. 키오스크 앞에 할아버지가 서 계신다. 아무래도, 주문을 못하고 계시는 것 같다. 얼른 도와드리고 우리 주문이 나오기를 기다렸다. 인스타에 사진을 찍어서 올린다. 여자친구가 폰을 불쑥 들이민다. '미친 놈들이 이런 거 보낸다니까!!' 외국 남성이 자신의 성기를 촬영해 전송했다. 얼마 전에는 사진을 도용당한 일도 있었다. 그럼에도 여자친구는 팔로워가 늘었다며 좋아한다. 여자친구가 유명하다는 카페에 가기로 했다. 문득, 요즘은 사진찍어야 한다고 서민들이 차린 카페는 오지도 않는다고 한탄하신게 생각났다.

오후 스토리

유발 하라리는 묘사한다. 현대의 사회를. '컴퓨터교'라고. 어쩌면 B.C는 Before Christ 가 아니라 Before Computer가 되야할지도 모르겠다. 아니 되야 한다. 최소한 컴퓨터교는 예수교 이상의 변화를 가져오고 있다. 우리는 컴퓨터에 의해 변하고 있다. 휴대전화의 출현으로 알람시계는 그 존재를 감추고 있으며, 손목 시계는 사치품으로 자리 잡았다. 인터넷은 사람들의 의견 표출을 자유롭게 했으나, 이로 인한 문제들도 상당수 발생하고 있다. 하늘 아래 좋기만 한 것이 있었던가. 우리, 흘겨보자. 두 눈을 길게 찢고, 컴퓨터 속 우리의 삶을 흘겨보자.

이부분은 직접적인 비판을 피할 것임. 위 글을 읽고 독자가 능동적으로 생각하도록

게임의 역습

1.

2.

3.

1. 게임이 뭐야

술게임, 온라인 게임, 콘솔 게임, 보드 게임. 세상에 게임은 많고 많다. 어디까지가 도대체 게임인걸까? 한국콘텐츠진흥원에 따르면 게임이란 '인간생활 중 노동을 제외한 여가생활에서 편을 가르거나 특정한 규칙을 세워 경쟁을 도입함으로써 유희적 재미를 배가한 레크리에이션의 일종' 이라고 나온다. 그런데, 우리가 원하는 것은 이런 고리타분한 것이 아니다. 정말. 무엇이 게임일까? 공통점을 찾아보자. 일단 즐겁다. 게임이 우리를 즐겁게 만들지 못한다면, 누가 그 게임을 하겠는가. 게임은 오락이다. 우리를 즐겁게하는데 제 1의 목표가 있다.

그러나, 즐겁기만 하면 게임인가? 즐겁기만 하다고 해서 게임이 아니다. 음악을 들어도 즐겁고, 맛있는 것을 먹어도 즐겁다. 심지어는, 사랑하는 이들과 함께 있기만 해도 즐겁다. 이것들이 게임인가? 음악 청취 게임(?), 미식 게임(?), 함께 있기 게임(?). 이것들이 게임일리 없다. 그렇다면, 무엇이 즐거운 것을 게임으로 만드는가.

위의 설명을 빌려보자. 일단 규칙이 있어야 한다. 위에 게임이 아니었던 것들은 모두 규칙이 없다. 그렇다면 규칙이 있고 즐거움에도 게임이 아닌 것은 무엇일까? 커뮤니티 활동이 있을 것 같다. 서로 지켜야 할 규칙은 있으면서도 즐겁다. 그러나 커뮤니티는 게임이 아니다. 왜일까. 인간을 넘어서는 규칙이 아니기 때문이다. 커뮤니티의 규칙은 커뮤니티를 원활하게 하는 일종의 편의를 위한 것이지 커뮤니티의 근본을 이루지 않는다. 그러나, 게임이라고 할 수 있는 축구에서 공을 손으로 잡고 달린다면, 그것은 아예 축구의 존재 자체를 무의미하게 한다. 그러므로 게임은, '규칙으로서 구현되는 즐거운 것'이라고 할 수 있다.

2. 게임의 대표 주자들

게임에는 무엇이 있을까. 우리가 흔히 즐기는 게임들을 살펴보자. 흔히 게임이라고 하면 온라인으로 즐길 수 있는 컴퓨터 게임과 오프라인에서 즐기는 게임을 말한다.

-컴퓨터 게임

대부분의 사람들이 게임 하면 떠올리는 것이 컴퓨터 게임일 것이다. 3줄 정도 간단한 소개

-스포츠

스포츠 또한 게임의 범주에 들어갈 것이다. 규칙에 의해 존재하고, 여가활동으로서 우리에게 즐거움을 준다.

-보드게임

보드게임 사진 첨부 예정

게임의 역습

게임만 한다고 엄마한테 등짝맞는 시대는 끝이 났다. 오락이었던 게임이 역습을 시작했다. 더이상 게임은, 우리를 즐겁게만 하지 않는다. 우리를 목적을 향해 한 발 짝 다가가게 해주려고 노력한다. 그런 게임을 총칭해서 기능성 게임이라고 한다. 게임의 정의와 마찬가지로 한국 콘텐츠 진흥원에 따르면 '게임적 속성을 갖추고, 목적지향성을 토대로 정보전달, 홍보, 인식 및 행동전환, 훈련 등을 목적으로 공공정책, 군사, 의료/건강, 교육, 기업 등의 분야에서 활용되는 게임'이 바로 기능성 게임이다.

기능성 게임의 종류

기능성 게임은 어떤 기능이냐와 그 정도에 따라 여러 종류로 나뉠 수 있을 것이다. 우선 사회적 기능과 개인적 기능으로 나뉠 수 있을 것이다. 또한, 사회적 기능성 게임은 크게 두가지로 나뉠 수 있는데 게임을 하고 그 게임의 광고 등으로 얻는 수익이 기능성을 띄는 경우. 이는 간접적인 방식으로 볼 수 있을 것이다. 그리고, 게임을 하는 그 과정 자체가 문제 해결인 경우를 직접적인 것으로 볼 수 있을 것이다. 그리고 마지막으로 교육용은 그 본질이 다른 기능성 게임과 다른데 문제 해결 과정이라기 보다는 게임 자체가 교육성을 띄는 경우이므로 논외로 한다.

예시

개인 기능성

해비티카

해비티카는 이용자들이 자신의 행동을 기록하고 목표 달성을 위한 동기부여를 할 수 있도록 게임처럼 만들어진 자기개발 웹 애플리케이션이다. 롤플레잉 게임의 형태를 띠고 있으며, 플레이어들은 골드나 갑옷을 모아 자신의 캐릭터를 더 강하게 만들 수 있다. 보상은 '습관'(Habits), '일일 과제'(Dailies), '할 일'(To-Dos)의 형태의 과제 수행을 통해 현실 세계의 목표 추구를 함으로써 얻을 수 있다.
-출처 위키피디아-

간접 기능성 게임의 예시

트리플래닛
나무 심는 게임.
리플래닛(Tree Planet)은 세상 모든 사람이 나무를 심을 수 있는 방법을 만드는 소셜 벤처이다. 개인 또는 그룹의 신청을 받아 숲을 조성하는 '크라우드펀딩 서비스'를 운영하고 있으며, '중국 사막화 방지숲', '세월호 기억의 숲', '네팔 지진 피해 지역 복구를 위한 커피나무 농장' 등 다양한 환경적, 사회적 가치가 있는 숲을 조성하고 있다. 또한, '소녀시대숲', '김수현숲' 등 전 세계 한류스타의 팬들이 숲을 만드는 스타숲 프로젝트를 통해 80여 개의 스타숲이 조성되었다. 2018년부터는 게임이나 멀리 있는 숲이 아닌 내 옆에 반려나무를 입양하면 숲이 필요한 곳에 나무를 심어주는, 강원도 산불피해 복구 숲, 미세먼지 방지 숲, 멸종위기종 보호 숲, 서울시 3천만 그루나무 심기, 개발도상국 자립형 커피나무 숲 등 캠페인을 진행하고 있다. 미세먼지에 취약한 아동, 어린이를 위한 초등학교 실내 교실 숲을 조성하고 있다.

게임에서 크라우드 펀딩으로 진화함.

직접 기능성 게임의 예시

폴드잇(자료)

이 게임을 통해 과학자들이 15년 동안 머리를 싸맸던 문제를 3주 만에 해결한 사례가 있다. 워싱턴대 데이비드 베이커 교수 연구팀은 원숭이에 에이즈를 일으키는 바이러스(M-PMV)의 단백질 구조를 밝혀내기 위해 한 게임을 만들었다. 해당 바이러스의 단백질 구조는 종류만 10만 개가 넘었기에 이들이 컴퓨터 계산으로 해독하기엔 벅찬던 것. 이에 연구팀은 2008년 온라인 게임 '폴드잇(Fold it)'을 개발했다. '폴드잇'은 일반 유저들이 단백질의 3차원 구조를 가상으로 만들어 볼 수 있는 게임으로 누구나 참여할 수 있었다. 사람들은 아미노산이 사슬들이 서로 엉켜있는 단백질 구조 안에서 사슬들을 이리저리 접으며 새로운 단백질을 만들었다. 게임에선 사람들이 더욱 안정적인 단백질의 구조를 만들 때마다 높은 점수를 부여했다.그 결과, 게임에 참여한 이용자들은 단 3주 만에 바이러스 구조를 판독했고 이에 2010년 8월 '온라인 게임을 통한 단백질 구조 예측'의 논문 공동 저자로 이름을 올렸다.

더 일상적으로 게임으로 바라보기

열정을 품은 타이머

허용된 앱 외에는 사용할 수 없는 규칙, 그리고 순수 공부시간만 측정한다는 규칙으로 서로의 공부시간을 가지고 경쟁할 수 있는 어플리케이션이다. 선풍적인 인기를 끌고 1등을 차지하기 위해 공부를 한다는 점에서 충분히 기능성을 띄는 게임이라고 볼 수 있을 것이다.

헌팅포차

헌팅포차에 가면 게임 단말기가 있다 남녀 간의 대화의 매개로 게임을 만들어 주는 것이다.

기능성 게임을 만들어 보자.

기능성 게임을 만들어 보자. 내가 만들 게임은 직접 기능성 게임이다. 사실 간접 기능성 게임은, 아무 게임이나 만들어 거기서 얻은 수익을 이용하면 어떻게든 만들 수 있기 때문에 구상해 보는 것이 의미가 없다.

기능성 게임의 구상

기능성 게임은 무엇보다, 사회 문제를 해결해야 한다. 그리고 게임의 본분에 충실해야 한다.

사회 문제들.

요즘 소개팅 앱들 과금 문제를 다른 방식으로 해결하고자 한다. 예를 들면 헌혈이라던가 봉사. 봉사를 게임으로 만드는 거다.

1안 번역 알바를 한컴타자연습에 도입하는 느낌으로 하자.

2안 메모리를 공유할 수 있나....CPU라던가....
ex) 리그오브레전드를 플레이하기 위해서는 다른 앱을 동시에 실행해야 함. 그런데 이 앱은 전 세계 컴퓨터들의 계산 능력을 분담해서 사용하는 문제를 푸는 거임.

3안 오류 분석하기. 게임에 접속하는 과정에서 한가지 오류 검증을 해야하고 이걸 통과하면 접속 됨. 통과 안되면 - 접속 안됨. - 어차피 다시켬. 사람들.

4 건축게임. 마인크래프트(디자인과 비용계산할 때 유용할 것으로 예상) 마인크래프트가 인기 많음. - 디자인을 건축가에게 파는 형식으로 하는 거지. 단가 계산도 자동으로 됨.

5. 개표과정을 게임으로 하면 안되나.....- 투표 개표를 게임으로

6. 게임을 이용한 커뮤니케이션

7. 미연시 연애 상담.

8. 익명으로 자신의 사진 올리면(비용 있음.) 다른 사람들이 거기다 화장하기 시작함. 그다음 순위를 정해서 좋아요 받음. 1등이 돈을 가져감. 꾸미지 못하는 사람들을 도와주는 역할.

3. 내 뇌가 컴퓨터라니?

나는 컴퓨터가 아니야!

컴퓨터는 기계, 나는 사람. 저놈은 내가 만든 것. 우리는 완전히 다르다. 아니 다른 줄 알았다.

실제 친구를 섭외하여 토론하고 그 토론 과정을 정리해서 담을 것임. 6월 1일에 예정되어 있음. 이것은 가르침보다 혹독한~과 비슷한 구성이 될 것으로 생각됨.

이외의 내용은 커넥톰, 뇌의 지도. 1.4킬로그램의 우주 뇌 등을 참고하여 작성 예정

어두운 태양

배경

멀지 않은 미래 삶에서 기업과 프로그래밍의 영향력이 매우 커짐
(아마존의 무인 유통, selling system / 정부의 자동화 행정 system)

많은 시스템이 자동적인 프로그래밍을 통해서 운영되고
"프로그래머" 의 자격에 기준선이 생김

어떠한 수준 "**웨이트 라인**" 이상의 프로그래밍에는 전문 자격증이 있어야 함.
이 전문 자격증을 어디서 제공하는지에 있어서 정부와 기업 간 알력다툼.

programming system을 제공하는 기업 "alpha"
수준 높은 프로그래머들로 이루어짐
알파는 기존의 기업들의 행태에 대해서 programming system을 제작하여 제공

기존의 research center를 갖고 있는 기업들도 alpha에게서 제공받는
programming system과 consulting system 의 경제성을 인정.
많은 기업들에 alpha의 프로그래머들이 포진되고 각 기업에서 지분을
키워가기 시작하며 무시하지 못할 group 으로 성장
alpha 에서는 자신들의 weight line 기준을 지정하고 입사시 test 로 활용하던
programming test를 기반으로 하는 새로운 'professional license system'을 도입.

생활수준 전반으로 programming의 영향력이 증대됨에 따라 정부에서
제공하는 다양한 행정 service의 부분에도 그 효율성을 위해
높은 수준의 programming으로 제공하는 service에 대한 필요성이 커짐

PERSONA

프로그래머 Alex

그의 친구 Vincent

선배 Rose

정부 요인 Garry

(1) Alpha의 프로그래머 Alex

alpha의 신입 개발자인 프로그래머 Alex는 Polynomial problem 과 NP 영역의 경계를 밝혀내는 연구를 하며 행정 시스템 개발팀에서 활동하고 있다. 빠르고 정확한 시퀀스가 생명인 행정 시스템 개발에서는 NP 영역의 문제를 사용하지 못하기 때문에, 그 경계에서 최대한의 효율을 만들어내는 연구가 필요하다. 많은 기업들에서 사람들의 일상생활에 깊게 관여하게 되다 보니 정부에서도 이러한 자동적인 프로그램을 활용하는 행정 시스템을 사람들의 생활 깊숙이 관여시켜야 하게 되었다.

특히 Alex는 대규모 보안 clouding system을 이용하여 사람들의 건강정보 공유와 관리 알고리즘을 담당하는 역할을 맡게 되었다. 전반적인 국민 건강정보 빅 데이터를 기반으로 사람들의 행동 양식과 삶의 습관을 통해서 유전적 정보를 통해 예측하는 것 보다 뛰어난 건강 예상 정보와 기능을 제공하는 “알고리즘”.

하지만 이는 형식상의 역할이었고, 실질적으로는 사람들의 생활 패턴과 행동 양식 그리고 사고의 흐름 등을 파악하여, 정치적 위험도가 높은 사람들을 걸러내고 기업과 정부에 필요한 / 불필요한 /위험한 사람들로 국민들을 나누어 관리하려는 목적이 뒤에 존재하고 있었다. Alex 는 이에 대해 알지 못하고 있었지만, 일련의 “사건”을 통해 이를 파악하게 된다.

(2) Alex의 숨겨진 사실

“사건” => Alex는 자신의 생활 패턴과 삶의 습관으로 알고리즘을 시험해 보기 위해 자신의 간단한 파일 정보를 찾아보는 와중, 다른 일반 파일들 보다 굳이 높은 보안 암호로 잠겨 있는 것을 확인하게 되고, 이를 의아하게 여겨 정부에서 일하고 있는 자신의 오랜 친구 Vincent를 설득하여 “실

질적 기밀"을 파악하게 된다. 정부와 alpha 간의 알력싸움에서 큰 부분을 차지하고 있던, user data 통계 알고리즘을 Vincent에게 흘리듯 넘겨주는 것으로 그의 도움을 받게 되는데... 그의 도움으로 자신의 파일을 열어 자신이 매우 위험과 매우 긍정(매우 도움) 그 두 개에 걸쳐있는 특이한 케이스의 사람임을 알게 된다.

이는 지금까지 alpha에서 만든 알고리즘을 통해 파악이 되어 있었으며, 지금까지 자신이 우연찮게 보고 듣고 했던 모든 생활이 모두 alpha에서 고의적으로 자신을 "매우 긍정"에 가깝게 만들기 위해 만들어낸 situation이라는 것을 파악한다. 이에 대해 프로그램을 지속해야 하는지에 대한 "갈등"하기 시작하는데, 이러한 situations들이 과연 자신에게 한정된 것인지 많은 시민들에게 이러한 의도적인 situations making이 이루어지고 있는지 파악하고자 함.

(3) Rose의 이야기

Alex는 그와 친한 선배 Rose를 찾아가 이에 대해 알고 있는 바가 있는지 물어봄. Rose와 대화를 해보다 보니 Rose는 이에 대해 이미 많은 것을 알고 있었다. 어떠한 방식으로 situation making이 이루어지는지, Alex가 alpha에 들어오게 되는 계기에 어떠한 방식으로 alpha와 정부가 개입을 했는지에 대한 것들 말이다.

"사실 Alex는 alpha에 입사하기 보다는 더욱 깊은 공부를

하고자 대학원 진학과 유학 등을 고려해보고 있던 찰나, 부모님이 큰 사고를 당하셨고 집안이 크게 기울었었다. 마침 alpha에서 좋은 제안으로 입사를 추천하여 alpha에 입사하게 된 일이 있었다."

Alex는 Rose가 어떻게 이러한 일들을 알고 있고, 언제부터 알고 있었는지에 대해서 캐묻기 시작했고 Rose는 자신도 예전에 이에 대해서 의아한 적이 있어서 파악을 하던 와중 알게 되었다고 밝힌다. Rose는 이러한 숨겨진 정부와 alpha의 음모를 더 이상 파헤치지 않고 순응해서 살아가는 것이 Alex에게도 도움이 될 것이라고 어필하며 타협하고 살아가자고 한다. 하지만 평소 즉흥적이고 불같은 성격의 Rose가 이렇게 쉽게 순응하고 살아가고 있다는 것을 의아하게 여긴 Alex는 Rose의 제안을 우선은 받아들이지만 뭔가 수상한 느낌을 받게 된다.

(4) Garry와 Rose

마침 Vincent에게서 추가적인 정보를 제공해준다는 연락을 받고 만날 약속을 잡은 Alex는 Vincent를 만나게 된다. Vincent는 Alex에게 지금까지 alpha와 정부 간의 알력다툼은 겉으로 그렇게 보여 지기 위한 연기와 같은 것이었고, 사실 정부와 alpha를 좌지우지하는 세력은 모두 한 통속이었다는 점을 알려준다. 이에 대해서는 Vincent도 새롭게 알게 된 사안이었고 Vincent도 이 음모에 대해서 더 이상 묵과할 수 없다고 생각해 Alex와 자세히 파악해보고자 한다.

Vincent와 Alex는 정부 요인 중 alpha와 깊게 연관되어

있는 사람을 찾고자 하는 부분이었고, 그들은 alpha의 공동 설립자들 중 한 명이 정부에서 새롭게 탄생시킨 부처인 “국민 건강관리 통계응용청”의 청장을 맡고 있다는 사실을 알게 된다. 그의 이름은 Garry Doll 이며, 그의 경력사항이 많은 부분 조작된 채 사이트에 올라와 있다는 사실을 알게 된다. Garry Doll은 자신이 alpha의 공동설립자라는 것을 숨긴 채 그저 오랜 기간 정부에서 일한 요인으로 소개하고 있었고, 이는 Alex와 Vincent에게 매우 수상하게 여겨졌다.

Garry Doll이 참가하는 정부의 국민 건강관리 홍보 세미나에 가서 Alex와 Vincent는 Garry Doll을 만나고자 한다. alpha의 주요 임원진과 몇몇 정부 요인들과 대화를 나누고 있는 Garry Doll에게 Vincent가 다가가 인사를 건넨다. 때마침 옆의 정부 요인이 Vincent에게 한 가지 팁을 무심코 전달하는데.. ‘Garry는 이름으로 불리는 것을 매우 싫어하니 Mr. Doll 이라고 말 하셔야 합니다.’
이에 대해 별 생각 없이 Vincent는 Alex에게 간단한 인사를 나눈 사실을 전달하게 되고, Alex는 뭔가 이상한 느낌을 받아 곰곰이 생각해 보는데.. Rose 의 full name이 Rose Doll 이었던 것이다. 당장 Rose에게 이에 대해 아는 것이 있는지 물어보려던 찰나 Vincent는 멈춰보라고 하고, 둘은 우선 Rose와 Garry의 관계에 대해서 찾아보았다.

알고 보니 Garry 와 Rose 는 부녀관계였던 것이다. 그렇다면 Rose가 Alex에게 했던 말들은 사실이었던 것인지에 대해 Alex는 고민을 하기 시작하고, 몇몇 정보를 더 취합하고자 한다. 집에 가서 무심히 자신의 사진첩을 바라보던

Alex의 눈에 어디선가 많이 본 사람이 자신의 학창시절 사진에 찍혀있던 것을 발견하고.. 자신의 대학 졸업사진 옆에 Rose가 서있는 것을 발견한다...

우리의 생각

(1) 사실 매우 도움과 매우 위험 둘 사이에 걸쳐있는 케이스는 아주 희귀한 케이스로 아직까지 Alex에게서만 나타나고 있었고, 이를 파악한 alpha와 정부 측에서는 Alex가 대학생이던 시절부터 그의 일과와 situations들을 관리하여 alpha의 보안 clouding system 팀으로 입사하게끔 만들어냈다. Rose는 Garry의 똑똑한 딸로 그 이전부터 Alex를 전담하는 agent로 활동했던 것이고 이를 Alex가 알아챈 것이다.

(2) Rose 또한, 매우 도움과 매우 위험 둘 사이에 걸쳐있는 케이스가 맞았고 자신이 Alex에게 말해준 것처럼 자신도 예전에 직접 알아낸 것이 맞지만, 뒤에 아주 많고 깊은 일들이 얽혀있다는 사실을 아버지를 통해 알게 되었고 이로 인해 Alex가 겪게 될 고난이 예상되어 순응하자고 했던 것이다.

=> Alex는 과연 어떤 선택을 하게 될까

시 사냥

[화창한 봄, 선선한 가을은 한창 새로운 곳으로 떠나는 말하

자면 이사철이다. 이사를 하며 느끼는 설렘과 떨림 새로운 것은 항상 떨리기 마련이다. 그 떨림 속에서 우리는 많은 것을 발견하고는 한다.]

박찬중의 '이사'를 감상해보자.

이사를 해보면 알지
오랜 세월, 참 많은
필요치 않은 것들을 끌고 다닌
허접한 잡동사니를 보게 되지
그럼에도 또 끊임없이
새로운 것들을 찾고, 그를 위해
애를 태우기도 하지

[우리는 이사를 하며, 많은 기억들 그리고 물건들, 그들이 우리와 함께했다는 것을 깨닫게 된다. 개중엔 분명 이제는 필요 없어진 것들도 있고 더 이상 도움이 되지 않는다고 여겨지는 것들도 있다. 하지만 분명 과거의 어디선가 에서는 훌륭한 나의 생각이었고 물건이었다. 지금 우리에게 컴퓨터가 있게끔 한 괴델과 튜링의 아이디어가 어떻게 나왔는지 생각해보면, 참 맞는 말이다.
1928년, 당대의 수학계는 꿈에 부풀어있었다. 기계적인 방식으로 - 자동으로 - 수학의 모든 사실을 술술 만들어낼 수 있다는 꿈 말이다.[1] 물론 이는 괴델에 의해서, 또 튜링에 의

1) 컴퓨터 과학이 여는 세계, 세상을 바꾼 컴퓨터, 소프트웨어 원천 아이디어 그리고 미래. 이광근 저, 인사이트. 2015. pg.28-29

해서 산산조각 난 꿈이 되어 버리고 만다. 만일 괴델과 튜링이 이사하면서 모든 것을 버리고 갔다면 어떻게 되었을까. 이제는 허접한 잡동사니가 된 과거의 영광을 보며 괴델과 튜링이 한 생각은 무엇이었을까.]

여기, 유치환의 '낙엽'에는 이런 구절이 있다.

"너의 추억을 나는 이렇게 쓸고 있다"

[세상에는 참 많은 우연찮은 일들이 있다. 어떻게 생각하면 필연인가 싶을 정도로 말이다. 그 일들은 서로 맞물려 새로운 이야기를 써내려가고, 부족한 과거를 메우고, 새로운 미래를 채워준다.]

이해인의 '아름다운 순간들' 중 일부이다.

마주한 친구의 얼굴 사이로,
빛나는 노을 사이로, 해 뜨는 아침 사이로...
바람은 우리들 세계의 공간이랑 공간은 모두 메꾸며...

[우연인지 필연인지 우리는 알지 못한다. 다만 매일매일 우리가 보고 듣는 모든 아름다운 순간들은 서로 메우고 연결

되고 같아진다. 1854년, 조지 부울은 <생각의 법칙에 대한 탐구>라는 책을 발표하며 사람의 생각은 조립되는 것이라고 본다. 그는 세 개의 접속사 그리고, 또는, 아닌 으로 사람의 생각을 충분히 조립할 수 있다고 보고 논리의 흐름을 나타내었다.[2] 마침 다른 곳에서는 디지털 스위치 회로가 기다리고 있었다. 부울의 논리를 만나기 위해 기다렸던 듯이, 두 생각의 세계는 정확히 일치하였고 새로운 이야기를 함께 써내려가게 되었다. 그것이 컴퓨터의 실현이라는 놀라운 이야기일 것이라고는 누구도 생각하지 못했을 것이다.]

[이는 고은의 '어떤 기쁨'을 떠올리게도 한다. 분명 지금 내가 생각한 것은 또 다른 어디선가 생각했고 생각하고 있는 것. 이는 이 자체로서 기쁜 일임에 분명하다. 어디선가 수많은 나가 새로운 이야기를 만들어 가고 있기 때문이다.]

지금 내가 생각하고 있는 것은
세계의 어디선가
누가 생각했던 것
울지 마라
...
얼마나 기쁜 일인가
이 세계에서
이 세계의 어디에서

---

2) 컴퓨터 과학이 여는 세계, 세상을 바꾼 컴퓨터, 소프트웨어 원천 아이디어 그리고 미래. 이광근 저, 인사이트. 2015. pg.54

나는 수많은 나로 이루어졌다
얼마나 기쁜 일인가
나는 수많은 남과 남으로 이루어졌다
[양자의 세상에는 독특한 세 가지 특징이 존재한다. 겹쳐있기와 얽혀있기 그리고 확률 진폭. 그 중 얽혀있기에 대해 말해보고자 한다. 양자들은 서로 짝을 이뤄 얽혀 있을 수 있다. 얽혀있는 두 양자 중 하나를 관찰해서 한 상태로 나타나는 순간, 얽혀있는 단짝 양자도 관찰된 상태와 짝을 이뤘던 상태로 나타난다.[3] 그 거리가 얼마나 되던 상관없이 단짝 양자와 맞물려 상태가 나타난다. 이 세상 어딘가에 같은 생각을 하는 수많은 나가 있다는 사실을 양자 알고리즘이 우리에게 알려주고 있다. 참 기쁘고 다행 아닌가.]

[나와 같은 누군가가, 함께 이야기를 만들어갈 누군가가 이 세상에 있다는 사실은 우리에게 큰 용기를 준다. 외롭지 않은 이야기를 함께 만들어나갈 누군가가 있다는 사실 말이다. 그러한 용기는 우리에게 새로움이라는 선물을 주곤 한다. 누구도 밟지 않았던, 가려 하지 않았던 길을 가는 자에게는 그러한 용기가 뒷받침 되었을 것이다. 여기 프로스트의 '가지 않은 길'을 감상해보자.]

노란 숲 속에 길이 두 갈래 갈라져 있었습니다.
안타깝게도 나는 두 길을 갈 수 없는
한 사람의 나그네로 오랫동안 서서

---

3) 컴퓨터 과학이 여는 세계, 세상을 바꾼 컴퓨터, 소프트웨어 원천 아이디어 그리고 미래. 이광근 저, 인사이트. 2015. pg.128 중 얽혀있기

한 길이 덤불 속으로 꺾여 내려간 데까지
바라다 볼 수 있는 데까지 멀리 보았습니다.

그리고 똑같이 아름다운 다른 길을 택했습니다.
그럴 만한 이유가 있었습니다. 거기에는
풀이 더 우거지고 사람이 걸을 자취가 적었습니다.
하지만 그 길을 걸음으로 하여
그 길도 거의 같아질 것입니다만,

그 날 아침 두 길에는 낙엽을 밟은 자취 적어
아무에게도 더렵혀지지 않은 채 묻혀 있었습니다.
아, 나는 뒷날을 위해 한 길은 남겨 두었습니다.
길은 다른 길에 이어져 끝이 없었으므로
내가 다시 여기 돌아올 것을 의심하면서.

훗날에, 훗날에 나는 어디에선가
한숨을 쉬며 이 이야기를 할 것입니다.
숲 속에 두 갈래 길이 갈라져 있었다고,
나는 사람이 적게 간 길을 택하였고,
그것으로 해서 모든 것이 달라졌다라고.

[우리는 살면서 그 누구도 가지 않은 길을 밟고자 할 때가 있다. 누군가가 갔던 잘 다져진 길이 우리를 유혹하지만, 한 번쯤 누구도 다듬지 않은 길을 다듬어 보고 싶다. 언젠가 누구도 가지 않았던 그 길도 다른 길들과 같아질 테지만, 지금 그 길을 가는 데에는 무엇보다 큰 용기가 필요하다. 그리고

이러한 용기의 걸음은 우리 세상의 많은 것을 바꾸어 놓는다. 이전에 튜링과 괴델이 산산조각 낸 꿈을 기억하는가. 당대 수학계를 이끌던 힐베르트는 새로운 도전을 하고자 한다. 기계적인 방식으로 – 자동으로 – 수학의 모든 사실을 술술 만들어낼 수 있다는 꿈 말이다.[4] 분명히 그는 새로운 길을 택해보고자 하였다. 당대 수학계를 이끌던, 모든 사람이 걷는 길을 단단히 다져놓은 그가 또 새로운 아름다움을 찾아 떠나본 것이다. 산산조각 났다고 비웃을 자가 있을까. 괴델과 튜링은 감사해야한다. 누구에게 감사해야할지는 우리 모두 알고 있다고 생각한다.]

[그렇게, 지금 우리에게 너무 익숙한 것으로 다가오는 것을 위하여, 그 뒤안길에는 많은 노력과 우연 그리고 필연이 존재한 것이다. 한 송이의 국화꽃을 피우기 위해 그 뒤안길에 수많은 노력과 우연 그리고 필연이 존재했던 것이다. 서정주의 '국화 옆에서' 중 일부이다.]

한 송이의 국화꽃을 피우기 위해
봄부터 소쩍새는
그렇게 울었나 보다.

한 송이의 국화꽃을 피우기 위해
천둥은 먹구름 속에서
또 그렇게 울었나 보다.

---

4) 컴퓨터 과학이 여는 세계, 세상을 바꾼 컴퓨터, 소프트웨어 원천 아이디어 그리고 미래. 이광근 저, 인사이트. 2015. pg.28-29

[무엇보다도, 지금까지의 생각처럼 새로운 이야기를 만들고 우리에게 중요한 혹은 정말 익숙한 것을 만들어내기 위해서는 어떻게 담아낼 것인지를 빼놓을 수 없다. 단지 생각만으로 모든 것이 이루어진다... 그것은 마법영화에서나 나오는 우리 모두의 바람일 뿐이다. 결국 우리는 써내려 가야하고 실현시키고 실행해야 한다. 이를 위해선 빼어난 그릇이 필요하다. 휘황찬란하고 거대한 그릇을 말하는 게 아니다. 합리적이고, 간결하며 우리와 잘 맞닿아 있는. 김시천의 '그릇'에 나와있듯이 말이다.]

그릇이 되고 싶다
마음 하나 넉넉히 담을 수 있는
투박한 모양의 질그릇이 되고 싶다
그리 오랜 옛날은 아니지만
새벽 별 맑게 흐르던 조선의 하늘
어머니 마음 닮은 정화수 물 한 그릇
그 물 한 그릇 무심히 담던
그런 그릇이 되고 싶다

1.1 튜링머신이 도대체 뭐야?

~~컴퓨터공학과를 전공하게 되면 . 그 중에서 앨런 튜링이라는 인물에 대해 관심이 많았던 사촌 동생과의 대화를 재구성 해 보겠다.~~

~~동생: 형, 얼마전에 튜링~을 봤는데 튜링기계가 도대체 뭐야?~~

~~나: 튜링기계는 앨런 튜링이라는 사람이 만든 기계야! 계산기 같은 거지.~~

~~동생: 근데 사람들은 왜 그렇게 튜링한테 관심이 많은거야?~~

폐기- 글 방향성이 너무 별로 여서 갈아 엎을 예정임.

2.1 암호화란?

우리는 살면서 어떤 누군가와 비밀스럽게 통신할 일이 생긴다. 예를 들어 준호에게만 내 시험 성적을 알려주고 싶다면, 나는 준호와 비밀 통신을 해야 한다. 준호와 내가 같은 공간에 있다면, 나는 준호에게 귓속말로 내 성적을 알려줄 수 있다. 하지만 준호와 내가 멀리 떨어져 있다면 나는 어떻게 내 시험성적을 준호에게만 알려줄 수 있을까? 이때 우리는 '암호화'라는 기술을 사용할 수 있다.

'암호화'의 정의는 매우 간단하다.

"보내고자 하는 메시지"를 "상대방만 이해할 수 있는 형식"으로 바꾸는 행위.

암호화는 항상 위의 정의를 따른다.

다시 똑같은 예를 들어 암호화의 정의를 확인해보자. 나와 준호는 어릴 때부터 같이 놀이터에서 비밀요원놀이를 해왔다. 비밀요원놀이의 규칙은 매우 간단하다. 놀이터에서 서로 숫자에 대한 내용을 이야기할 때는 항상 그 숫자에 2를 곱하고 3을 더한 후 이야기하는 것이다. 이제 준호와 나는 서로만 이해할 수 있는 수단이 생겼다. 〈숫자를 이야기할 때 2를 곱하고 3을 더한다〉. 서로만 이해할 수 있는 수단이 있으니 우리는 이제 암호화를 진행할 수 있다. 나는 내 성적- 95점을 준호에게 비밀스럽게 알려주고 싶다. 이때 공유하고자 하는 내용- 내 성적이, 바로 '보내고자 하는 메시지'다. 나는 내 성적 95점에 2를 곱하고 3을 더한다- 193. 이 과정이 바로 '상대방만 이해할 수 있는 형식'으로 바꾸는 과정이다. 이 193이라는 숫자를 편지지에 적어 준호에게 보낸다. 편지가 전달되는 도중에 같은 반 친

구 유진이가 편지를 가로채서 내용을 읽어봐도 유진이는 이 193이라는 숫자가 무엇을 의미하는지 알 수 없을 것이다. 나와 준호만 알고 있는 수단으로 내용을 '암호화' 했기 때문이다.

이처럼 우리는 평소 일상에서 상대방과 나만 알 수 있는 내용을 공유한다. 메신저로 친구에게 어제 데이트는 어땠는지, 소개팅 상대가 마음에 들었는지 물어본다. 이때 우리는 너무나도 당연하게 이 내용이 나와 내 친구 사이에서만 볼 수 있다고 생각한다. 메신저를 통해 문자를 보내는 행동도 모두 '암호화' 과정을 거치는 것이다.

이제 암호화에 대한 간단한 정의와 사용 용도를 알게 되었으니, 암호화의 과정에 대해 더 자세하게 알아보자. 암호화라는 기술을 이해하기 위해선 알아야하는 중요한 개념이 몇 가지 있다. 바로 '송신인/ 수신인', '평문/ 암호문', '암호화/ 복호화' 그리고 '열쇠 (키)'에 대한 개념이다.

'송신인'은 비밀스러운 내용을 보내는 대상, '수신인'은 해당 내용을 받으려고 하는 대상이다.

'평문'은 암호화가 진행되기전의 원본 메시지이고, '암호문'은 암호화가 진행된 후의 결과물이다.

'암호화'는 평문을 암호 알고리즘을 통해 이해하기 어려운 암호문으로 바꾸는 과정을 칭하고, '복호화'는 암호문를 다시 평문으로 바꾸는 과정을 칭한다.

'열쇠 (키)'는 암호화와 복호화에 사용되는 도구다. 열쇠를 사용하여 암호문을 만들고, 열쇠를 사용하여 암호문을 평문으로 돌려낸다.

우리는 위의 개념들을 사용하여 암호화의 네 단계를 서술할 수 있다.

1. '송신인'은 암호화 '열쇠 (키)'와 알고리즘을 사용하여 '평문'을 '암호화'한다.

2. '암호화'된 '암호문'을 '수신인'에게 전달한다.

3. '수신인'은 복호화 '열쇠 (키)'와 알고리즘을 사용하여 '암호문'을 '평문'으로 '복호화'한다.

4. '수신인'은 '평문'을 읽는다.

위의 단계를 고스란히 우리의 예제에 적용해보도록 하자.

1. '나'는 암호화 열쇠: 'x=2, y=3'과 알고리즘: '메시지*x + y'를 사용하여 '성적: 95'를 '암호화'한다. (95*2 + 3 = 193)

2. '암호화'된 '성적: 193'을 '준호'에게 전달한다.

3. '준호'는 암호화 열쇠: 'x=2, y=3'과 알고리즘: '(메시지-y) / x'를 사용하여 '성적:193'을 '복호화'한다. ((193-3) / 2 = 95)

4. 이제 '준호'는 '성적: 95'를 읽는다. 준호는 다른 사람이 모르게 나의 성적을 알게 된 것이다!

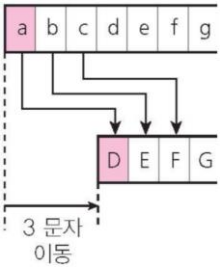
과연 암호화의 개념은 이게 끝일까? 단순히 나와 상대방이 서로만 아는 하나의 규칙이 있다면 나는 이 세상 모든 것을 상대방과 안전하게 공유할 수 있을까?

암호화의 세계는 그렇게 호락호락하지 않다. 암호 작성자들은 더욱 어려운 암호를 만들고 암호 해독자들은 암호를 풀 수 있는 강력한 방법을 열심히 찾아낸다. 이제 암호화의 역사를 알아보며 암호화의 종류와 이의 발전을 알아보도록 하자.

2.2 고전 암호와 암호의 기계화

암호화의 역사는 다른 학문과 비교해도 상당히 길다. 암호화는 기원전 404년, 스파르타 시절 때부터 사용된 것으로 알려져 있다. 하지만 암호화의 실체가 공식적으로 드러난 것은 기원 후 2세기경에 세토니우스가 쓴 'Lives of the Caesars'에서이다. 이 책에선 로마의 장군- 율리우스 카이사르가 사용한 암호화 방법에 대해 자세하게 다룬다.

율리우스 카이사르가 사용한 암호를 우리는 '카이사르 (시저) 암호'라고 부른다. 카이사르 암호는 치환 암호 (substitution cipher) 중 하나이다. 카이사르는 메시지를 전달할 때, 각각의 글자를 항상 해당 글자의 세 자리 뒤에 오는 알파벳으로 치환했다. 카이사르 암호를 사용해 평문 알파벳과 암호화된 알파벳을 비교해보면 아래와 같아질 것이다.

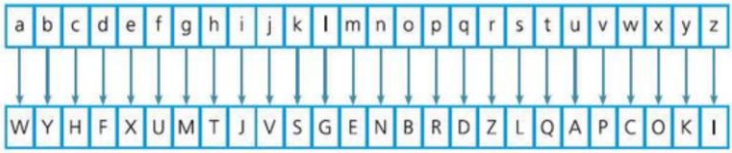


카이사르 암호를 직접 사용해보도록 하자. 예를 들어 단어- 'add'를 카이사르 암호를 이용해서 암호화하고자 한다. 그렇다면 우리는 a의 치환 값 D, d의 치환 값 G를 사용하여 암호문 'DDG'를 만들어 낼 것이다. 이제 이 'DGG'를 상대방에게 전달하면 상대방은 D의 세 자리 앞에 오는 알파벳 a, G의 세 자리 앞에 오는 알파벳 d를 찾아내므로 암호문을 평문- 'add'로 다시 복호화 하여 읽는다.

카이사르 암호 역시 암호화의 기본 규칙을 따른다. 카이사르 암호의 경우, 암호 알고리즘은– '각 알파벳을 어떤 특정 알파벳으로 치환한다'이다. 열쇠는 '치환 알파벳은 해당 알파벳의 3자리 뒤에 오는 알파벳'이라는 정보다. 송신인은 해당 알고리즘과 키를 사용하여 평문을 암호화하고, 수신인 역시 해당 알고리즘과 키를 사용하여 암호문을 평문으로 복호화한다.

카이사르 암호는 알파벳의 3자리 평행이동만 고려하는 치환 암호이지만, 더 많은 자릿수 이동 (알파벳의 경우 최대 25번) 그리고 알파벳의 재배열까지 허용한다면 400,000,000,000,000,000,000,000,000가지의 서로 다른 열쇠가 생성될 수 있다.

아래는 재배열을 사용한 치환 암호이다.



이 경우 단어 'add'를 암호화하면 암호문– 'wff'을 얻을 것이다.

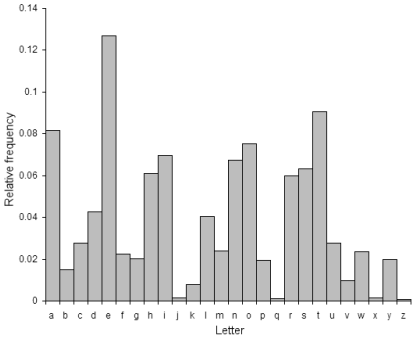
400,000,000,000,000,000,000,000,000가지의 경우의 수가 있다면 이 암호는 안전한 것이 아닌가? 그렇다면 우리는 지금도 기원 전에 만들어진 암호화 기법을 사용하고 있는 것 인가?

안타깝게도 대답은 '아니요'다. 율리우스 카이사르와 같이 암호를 만드는 암호 작성자들이 있다면, 그 뒤에선 암호를 풀려고 노력하는 암호 해독자들의 성과가 있다. 암호 해독자들은 치환

암호를 해독하는 방법을 찾아냈다.

암호 해독자들은 언어의 성질을 이용해서 치환 암호를 해독해냈다. 이 때 이용된 언어의 성질이란 언어에서 나타나는 특정 글자의 빈도수를 칭한다.

우리 앞에 있는 영어 책의 한 페이지를 펼치고, 무작정 그 페이지에 있는 알파벳(a-z)의 각각 빈도수를 세 보자. 알파벳의 빈도수를 그래프로 나열하면 아래와 같은 형태가 나타날 것이다.



이렇게 빈도수 분석을 진행해보면, 영어에선 알파벳 e가 제일 많이 반복되어 사용되고 t와 a를 그 뒤를 잇는다. 우리는 바로 이 빈도수 분석을 사용해서 치환 암호를 해독할 수 있다.

예를 들어 어떤 임의의 치환 암호를 사용한 암호문의 알파벳 빈도수를 측정하니 m의 빈도수가 제일 높은 것으로 나왔다. 이 경우, 우리는 e가 m으로 치환되었다는 합리적인 예상을 할 수 있다. 또, 암호문에서 m다음으로 b와 f의 빈도수가 높은 것으로 나왔다면, t와a가 b또는f로 치환되었다는 예상을 할 수 있

다.

하지만 빈도수가 비슷한 알파벳의 경우 우리는 암호 해독에 실패할 수 있다. 예를 들어 영어에서 g, p, y의 빈도수는 매우 비슷하다. 암호문에서도 a, j, l의 빈도수가 매우 비슷하다면, g가 a로 치환됐는지 j로 치환됐는지, l로 치환됐는지 알 수 없다. g는 a 또는 j 또는 l이다 정도의 예상만 할 수 있다.

암호 해독자들은 이와 같은 경우에 대해서도 준비가 되어 있었다. 암호 해독자들은 언어의 또 다른 특징을 사용한다. 바로 특정 알파벳이 어떤 다른 알파벳 앞 또는 뒤에 많이 나타나는지를 조사하는 것이다. 예를 들어 n이라는 알파벳 앞에는 모음(vowel- a, e, i, o, u)가 많이 나타나는 경향을 보인다. 빈도 분석에 이 정보를 고려한다면 더욱 쉽게 암호를 해독할 수 있을 것이다.

또한, 언어마다 반복되는 단어가 있는 경우가 많다. 예를 들어 영어에선 and, the, a, an과 같은 단어가 많이 반복된다. 암호문에서도 반복적으로 나오는 세 글자가 있다면 이는 평문의 and 또는 the일 확률이 높을 것이다.

우리는 빈도수 분석, 알파벳 앞 뒤에 어떤 다른 알파벳이 많이 나타나는 지, 어떤 단어들이 많이 반복되는 지 등에 대한 정보를 통해 치환암호를 복호화 (해독)할 수 있다. 이는 치환암호는 안전하지 않다는 것을 알려준다. 누구나 마음먹고 위에 분석 방법을 사용하여 암호문을 평문으로 돌려낼 수 있기 때문이다. 분석 방법이 매우 쉬어 요즘은 치환암호 복호화 웹사이트를 통해 5초만에 치환 암호문을 평문으로 복호화 해낼 수 있다.[3]

세계대전 당시 독일 군사 지휘부는 중요한 군사명령을 독일이 아닌 다른 지역에 있는 군인들에게 비밀스럽게 전달해야 했다. 군사명령이 연합군 손안에 들어가면 작전이 모두 수포로 돌아가기 때문에 치환암호같이 해독이 쉬운 암호기법은 사용할 수 없었다. 대신, 독일군은 '에니그마'라는 암호 기계를 통해 중요한 명령들을 암호화하여 통신할 수 있었다.

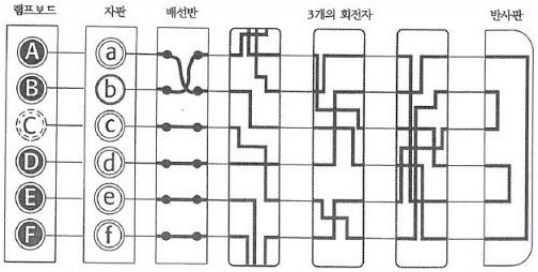
에니그마는 암호 기계로, 1918년 독일의 발명가 아르투르 세르비우스가 발명했다. 에니그마 기계는 크게 세 가지 부분으로 이루어져 있다. 평문을 입력할 자판 (키보드), 평문을 암호문으로 바꿔주는 변환기, 암호문을 보여줄 램프보드로 이루어져 있다.

위에 볼록 올라온 키보드로 원하는 글자를 누르면, 기계 안에서 암호화 과정이 진행되고, 암호화된 글자가 램프보드에 표시된다. 예를 들어 B를 눌렀는데 G에 불이 들어온다면, B가 G로 암호화된 것이다.

치환 암호의 경우 어느 특정 글자는 항상 똑같은 글자로 치환된다. 예를 들어 A가 G로 치환된다는 암호 규칙이 있다면 평문에서 등장하는 모든 A는 G로 치환될 것이다. 하지만 에니그마 기계의 경우, 글자를 암호화할 때마다 그 글자에 대응하는

글자가 항상 달라진다. 평문에서의 A가 에니그마 암호화 과정을 거치면 C가 될 수도, N이 될 수도, U가 될 수도 있다. 에니그마 암호 변환기는 매우 정교하게 설계되어 있어 글자가 키보드로 입력될 때 마다, 항상 자기 자신이 아닌 임의의 글자로 암호화된다.

에니그마 내의 암호 변환기는 배선반, 3개의 회전자 그리고 반사판으로 이루어져 있다. 에니그마 암호 변환의 원리는 일반인이 이해하기에는 한계가 있다. 첫째로, 전기와 전류에 대한 개념을 알아야 한다. 둘째로, 배선반을 사용해 글자의 전류 소켓들을 바꿀 수 있다는 점을 이해해야 한다. 셋째로, 3개의 회전자가 어떻게 작동하는지, 회전자의 시작위치와 배열 순서는 어떻게 영향을 미치는지 알아야 한다. 네번째로, 반사판의 원리와 에니그마 기계에 어떻게 암호화된 결과가 램프보드에 표시되는지 알아야한다. 이 네 가지 사실을 사용하여 아래와 같이 글자를 암호화 할 수 있다.



위의 네 가지 내용과 작동 그림을 보기만 해도 어려워 보이고 싫증이 난다. 에니그마 암호 변환의 작동 방식은 매우 흥미로운 것은 맞으나, 우리에게 중요한 것은 작동 방식이 아니라 에니그마라는 기계가 암호학계에 미친 영향이다. 에니그마의 자

세한 작동원리를 알고 싶은 독자가 있다면 데이비드 칸의 〈에니그마 입수작전〉이라는 책을 읽어보는 것을 추천한다.

에니그마는 암호 작성자들의 엄청난 성과이고, 암호 해독자들에게는 매우 골치 아픈 존재이다. 에니그마가 골치 아픈 이유는 가능한 열쇠 (키)의 개수 때문이다. 에니그마 사용자 (송신자/ 수신자)에게 필요한 열쇠 (키)는 회전자의 초기 위치, 3개 회전자의 배열 순서, 배선반 연결 방식- 이렇게 총 3가지이다. 그렇다면 열쇠 (키)의 총 경우의 수는 얼마일까? 에니그마의 회전자 위치의 경우의 수 17,576가지, 회전자 배열의 경우의 수 6개, 배선반 연결하는 방법의 수 100,391,791,500가지- 총 17,576 * 6 * 100,391,791,500가지의 경우의 수가 존재한다. 이는 10명의 사람이 24시간 동안 1년 내내 쉬지 않고 계산해도 암호를 푸는 대 약 2000만년정도의 시간이 걸린다. 사실상 에니그마 암호는 해독 불가인 것으로 보인다!

당시 독일군과 맞서 싸우던 연합군도 에니그마의 위력에 낙담했고, 독일군도 에니그마가 절대 풀리지 않을 것이라고 호언장담했다.

하지만 이렇게 대단해 보이던 에니그마 암호도 암호 해독자들에게 당하고 만다. 영국 암호 해독반 40호실과 블레츨리 파크에서 활동하던 앨런 튜링이 에니그마 암호를 해독할 수 있는 봄브 (The Bombe)라는 기계를 만들어낸 것이다.

위의 거대한 기계가 바로 '봄브'이다. 봄브를 발명한 공은 모두 앨런 튜링에게 고스란히 갔지만, 사실 폴란드의 레예프스키가 위와 비슷한 일을 하는 기계를 미리 발명해냈다. 또한, 에니그마의 작동원리도 모르던 연합군에게 에니그마 기계 지침서를 제공해준 독일인 한스 틸로 슈미트의 공도 있다. 에니그마 기계 지침서를 통해 연합군은 에니그마 기계를 그대로 만들어 볼 수 있었다. 연합군은 이 에니그마 기계와 독일군이 버리고 간 코드북 (1달 동안, 매일 사용될 열쇠가 기록 되어 있는 책) 그리고 반복되어 사용된 키들의 규칙을 이용하여 봄브를 만들어낸 것이다.

앨런 튜링이 겪은 난관, 봄브를 개발하고 발명할 때 이용된 기술적인 세부사항은 모두 생략한다. 에니그마의 암호화 원리를 보면서 알았겠지만 이 기계는 이해하기 쉬운 존재가 아니다. 우리가 에니그마를 통해 알아가야 할 중요한 것은 바로– 정교하게 기계화된 암호 역시, 사람이 풀 수 있다는 점이다.

2.3 현대 암호학- 공개 키 암호

암호화 기법에는 두 가지 종류가 있다. 대칭 키 암호화 방식과 비대칭 키 암호화 방식이다.

일단 대칭 키 암호화 방식에 대해 알아보자. 대칭 키 암호화 방식에선 암호화에 사용되는 열쇠 (키)가 복호화 할 때 사용되는 열쇠 (키)와 동일하다. 우리가 책에서 이때까지 다룬 암호화 기법들은 모두 대칭 키 암호화 방식이다. 카이사르 암호의 경우 '3번 평행이동 했다' 라는 하나의 동일한 열쇠가 암호화와 복호화 과정에 사용된다. 에니그마 암호의 경우도 동일한 코드북 (매일 사용될 열쇠를 포함한 책)을 사용하여 암호화와 복호화 과정이 진행된다.

그렇다면 비대칭 암호화는 무엇일까? 비대칭 암호화 방식에선 암호화할 때 사용되는 열쇠 (키)가 복호화 할 때 사용되는 열쇠 (키)와 다르다. 비대칭 암호화 기법은 공개키 암호화 방식이라고 불리기도 한다. 공개키 암호화 방식이 도대체 무엇을 의미하는가?

앨리스와 밥이 공개키 암호화 방식으로 소통한다고 가정해보자. 앨리스와 밥은 아래의 단계를 밟을 것이다.

1. 송신인인 앨리스(A)는 공개키 (public key)와 개인키 (private key)를 생성한다. A는 자신의 공개키를 모두에게 공개한다.

2. 수신인인 밥(B)도 공개키 (public key)와 개인키 (private key)를 생성한다. B는 자신의 공개키를 모두에게 공개한다.

3. 송신인 앨리스 (A)는 수신인 밥 (B)의 공개키 (public key)를 사용하여 보내고자 하는 메시지를 암호화한다.

4. 수신인 밥 (B)는 자기 자신의 개인키 (private key)를 사용하여 받은 암호문을 평문으로 복호화한다.

독자들은 의아해할 수 있다. 열쇠 (키)가 공개되면 아무나 다 암호문을 열어볼 수 있는 게 아닌가? 공개키를 세상 모두에게 공개한다는 게 무슨 의미인가?

이해하기 쉬운 예제를 들어보도록 하겠다. 앨리스 (A)는 금고상자를 밥 (B)에게 보내고 싶다. 앨리스는 직접 자물쇠와 그에 맞는 열쇠 하나를 만든다. 이때 앨리스의 자물쇠는 공개 키의 역할을 하고 자물쇠에 맞는 열쇠는 개인키의 역할을 한다. 밥도 직접 자물쇠와 그에 맞는 열쇠 하나를 만든다. 밥의 경우도 자물쇠가 공개 키 역할을 하고 열쇠가 개인키의 역할을 한다. 수신인인 밥은 자신의 자물쇠를 모두가 쓸 수 있는 곳에 걸어둔다. 앨리스는 자신의 금고상자를 밥의 자물쇠 (밥의 공개키)를 이용하여 잠근다. 이제 금고상자에는 밥의 자물쇠 (밥의 공개키)가 채워져 있다. 밥은 앨리스에게서 금고상자를 전달받는다. 밥은 자신의 열쇠 (밥의 개인키)를 이용하여 금고상자를 연다.

이게 바로 공개키 암호화의 방식이다. 밥의 자물쇠로 금고상자를 잠갔다면 그 누가 중간에 금고상자를 가로채도 밥의 자물쇠를 열지 못할 것이다. 밥의 자물쇠에 맞는 열쇠를 가지고 있지 않기 때문이다!

현재 우리가 온라인으로 주고받는 문자나 중요한 이메일 내용들 모두 이 공개키 암호화 방식을 사용한다. 자물쇠와 열쇠를 사용한 예제는 이해하기 쉬웠다. 하지만 온라인으로는 자물쇠라는 물건과 열쇠라는 물건이 존재하지 않는데 도대체 어떻게

소통한다는 것인가? 온라인에서 비대칭 암호 기법으로는 RSA 라는 암호체계가 사용된다. RSA 암호화체계는 소인수분해의 개념을 사용한다.

컴퓨터 과학과 수학계에선 P와 NP 문제라는 개념이 존재한다. P문제는 '풀기 쉬운 문제'로 다항 비용의 알고리즘을 가지고 있는 문제다. P문제는 컴퓨터로 현실적인 시간안에 풀 수 있다. NP문제는 '운에 기대면 현실적으로 풀 수 있는 문제'로 다항식 비용 알고리즘이 있는지 없는지도 알 수 없다. 우리는 NP 문제를 대게 어려운 문제라고 칭하며, 컴퓨터로 풀 수 없다고 가정한다.

소인수분해는 NP문제다. 다항식 비용으로 소인수 분해를 할 수 있는 알고리즘을 우리는 아직 찾지 못했다. RSA는 바로 소인수분해의 이 성질을 사용해 공개키 암호화를 실현해낸다.

이제부터 RSA 알고리즘에 대해서 자세히 알아보자. 이제부턴 설명이 어렵더라도 천천히 따라와 보도록 하자.

1) 앨리스와 밥이 소통하고자 한다. 앨리스는 값이 매우 큰 소수 p와 q 값을 정한다. 앨리스가 p값으로 17, q값으로 11을 고른다고 가정하자.

2) 앨리스는 p와 q를 곱한 값 N을 구한다. 이 경우, N값은 17x11 = 187이다. 앨리스는 이제 또 다른 새로운 값 u를 고른다. 이 경우, 앨리스가 u값으로 7를 고른다고 하자.

3) 앨리스는 모두가 볼 수 있는 공간에 자신이 가지고 있는 N값과 u값을 공개한다. 여기서 N&u값이 바로 앨리스의 공개 키이다.

4) 앨리스는 자신이 보내고자 하는 메시지를 숫자로 일단 변환한다. ASCII 코드 또는 UTF-8 코드 등으로 메시지를 숫자로 변환할 수 있다. 이 숫자를 M이라고 하자. 우리는 이제 이 M을 암호화할 것이다. 아래의 공식을 이용해 암

호문 C를 만들어 낸다.

$$C=M^u(\text{mod } N)$$

5) 밥이 앨리스에게 보내려고 하는 메시지를 ASCII 코드로 변환한 값이 88이라고 하자(M=88). 밥은 모두가 볼 수 있는 공간에서 앨리스의 N값과 u값을 찾는다. (N=187, u=7). 위의 식에 88, 187, 7를 대입하면 C값으로 11이 나오게 된다.

$$C=88^7(\text{mod}187)=11(\text{mod } 187)$$

6) 밥은 이 암호문 C=11을 앨리스에게 전달한다. 이때 악당 이브가 앨리스에게 전달되는 메시지를 중간에 낚아챘다고 가정하자. 이브는 11이라는 숫자만으로 원래 메시지 값 88을 구할 수 없다. 지수 모듈러 연산은 일방향 함수이기 때문이다.

7) 앨리스는 밥에게서 받은 C (11)를 복호화 할 수 있다. 앨리스는 p와q값을 가지고 있기 때문이다. 앨리스의 개인키 d는 아래의 공식을 통해 구할 수 있다.

$$u \times d = 1(\text{mod } (p-1) \times (q-1))$$

우리의 경우, $7 \times d = 1(\text{mod } 16 \times 10) = 1(\text{mod } 160)$로 23을 d값으로 얻게 될 것이다. 앨리스는 이제 이 d값을 이용해 밥에게서 전달 받은 암호문 C를 복호화 할 수 있다.

8) 앨리스는 아래의 연산을 통해 원래 메시지 M값을 구한다.

$$M=c^d(\text{mod}187)$$

우리의 경우, $M=11^{23}(\text{mod}187)$로 M=88이라는 값을 얻게 된다. 이는 밥이 앨리스에게 처음 보내려고 한 M=88과 같다! 앨리스는 밥의 암호문을 해독해 낸 것이다!

위의 RSA 알고리즘 단계들은 모두 일방향 함수- '두 개의 큰 소수 p와q로 N값을 구하는 건 쉽지만, N값으로부터 원래의 소수 p와 q를 찾는 것은 어렵다'를 활용한다. 이 일방향 함수

가 NP문제의 역할을 하는 것이다.

온라인으로 공유되는 데이터들은 모두 이 RSA 알고리즘을 사용한다. 사실 디지털 서명이라는 기술이 있어야 온라인의 통신을 완벽하게 이해할 수 있지만, 우리는 여기서 만족하도록 하자.

2.4 이제는 무엇을 해야 될까?

이 세상에는 매우 다양한 암호화 기법이 존재한다. 카이사르 암호, 동음 단일 치환 암호, 비즈네르 암호, 돼지우리 암호, 플레이페어 암호, ADFGVX암호 등 모두 암호화 기법 중 하나다. 암호 작성자들은 매번 풀리지 않는 암호를 만들려고 노력한다. 암호 해독자들은 매번 암호를 풀기 위해 노력한다.

근 몇 천년의 과거를 본다면 승리는 대부분 암호 해독자들에게 돌아갔다. 1586년도 비즈네르 암호가 나왔을 때, 아무도 비즈네르 암호를 풀 수 없다고 생각했다. 하지만, 암호 해독자들은 비즈네르 암호를 풀었다. 세계 대전 당시 독일이 활용한 에니그마 (기계화된 암호) 역시 아무도 풀 수 없다고 생각했다. 하지만, 암호 해독자들은 에니그마 암호 역시 풀어냈다.

지금은 다시 암호 작성자들의 승세다. 암호 작성자들은 RSA알고리즘, D-H알고리즘, ECC알고리즘 등 암호 해독자들이 풀지 못하는 암호화 기법들을 만들어냈다. 암호 해독자들은 이 암호화 기법들을 해독하려고 노력 중이다. 현재 활발히 개발되고 있는 양자 컴퓨터 역시 암호 해독자들의 움직임으로 봐도 무방하다. 양자 컴퓨터를 적절히 활용하면 NP문제를 다항 시간 내에 풀어낼 수 있다. 자연의 힘을 빌려 양자 컴퓨터는 어려운 문제를 무지하게 빠른 시간안에 풀어낸다.

암호 작성자들은 다시 긴장 중이다. 이 세상 어느 누군가가 소인수분해를 푸는 알고리즘을 발견해낸다면 당장 우리가 사용하고 있는 RSA암호가 무용지물이 될 수 있다. 암호 작성자들은 RSA와 ECC 같은 암호 기법에는 더 이상 안중에 없다. 양자 컴퓨팅의 발전과 동시에 암호 작성자들은 '포스트 양자 암

호’의 개발을 시작하고 있다. 격자 암호, 해시 기반 암호 등 효율적인 완전동형 암호를 개발하는 것이 암호 작성자들의 현재 목표이다.

암호화는 우리 삶에 매우 필요한 존재이다. 핸드폰으로 안전하게 계좌이체를 하고 민감한 회사 내용을 이메일로 소통할 수 있는 것도 다 ‘암호화’ 덕분이다. 우리는 이 책에서 암호화의 개념과 역사를 살펴보았고, 현재 우리가 어떻게 안전한 통신을 하는지 알아봤다. 암호학계에선 언제나 암호 작성자와 암호 해독자들의 싸움이 벌어지고 있다는 점을 독자들이 기억해 줬으면 한다.

3.1 우리는 왜 페이스북에 열광했나?

페이스북은 수년간 ~~

페이스북은 자료 조사만 진행하였음.

페이스북이 사용하는 알고리즘을 이해하는데 시간이 오래 걸려 이를 글로 녹여 내기 위해선 시간이 더 걸릴 듯.

페이스북-케임브리지 애널리티카 정보 유출 사건과 관련해선 배경지식이 더 필요할 것 같아서 넷플릭스 다큐멘터리〈The Great Hack〉을 시청 중.

이번주 안으로 다큐멘터리 시청과 자료조사를 마무리 한 후 글 작성 예정.

참고자료 (정리 안됨)

https://jhnyang.tistory.com/187

https://m.blog.naver.com/koromoon/220568499693

3- http://www.chaos.org.uk/~eddy/craft/substitute.html

http://home.bt.com/tech-gadgets/cracking-the-enigma-code-how-turings-bombe-turned-the-tide-of-wwii-11363990654704