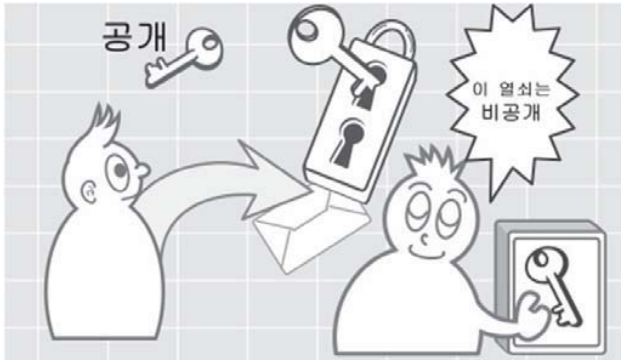
정보 윤리 소책자

공개 열쇠 암호는 숨은 공로자

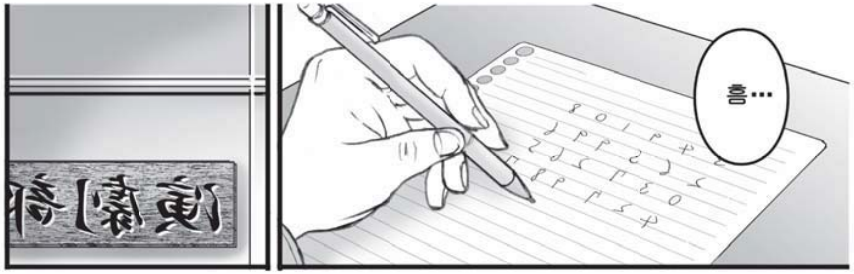


【 목표와 포인트】

1) 대칭 암호와 공개 키 암호화의 차이점과 특징에 대해 이해한다.
   - 공개키 암호화는 어떤 암호 시스템인지 이해한다.
   - 공개키와 비밀키가 어떤 장면에서 사용하는 방법을 이해한다.

2) 암호 통신 및 전자 서명의 관계를 이해한다.
   - 인증 기관 (CA : Certificate Authority)의 역할을 이해한다.
   - 공개 키 인증서가 무엇인지 이해한다.
   - SSL (https://) 통신의 처리를 구체적으로 이미지 할 수 있게 된다.
   - 구체적으로 인증 기관의 전자 서명을 확인하고 공개 키 인증서가 수정되지 않았거나 등을 확인된다.

3) RSA 암호의 구조와 안전성에 대해 이해한다.


흠…


뭐야~
벌써
포기?
3분내에
못 풀면
벌칙이야
자..잠깐
기달려 주세요.


아까부터
뭘 그렇게
하고 있어?


아오야마상이
만든
암호를 풀고
있는데..


리카, 너 암호를
얕보지 말라고!
암호 없이 지금의
인터넷 사회도
성립하지 않는다고!
암호에
웃는자
암호에
운다!


암호?
그거
재미있는
거야?...


그건 분명 암호를
사용하는 거라고.
공개키라던가,
EMI라고 하는…
예를 들어,
패스워드 같은거
보낼 때, 열쇠마크가
표시되지?
뭐야 그게??
무슨 소리인지
잘 모르겠는데


그보다 EMI는
레코드 회사
아닌가?
뭔가요?
그 공개키
라는게?


뭐, 어쨌든
국가에서도 암호를
사용해서 전자정부
같은 서비스를
시작할 정도로
암호는 중요한
거야
으…
수업 좀 성실하게
들어뒀어야 했는데
…


열쇠를
공개하는 거야?
위험하지
않아?
근데, 공개키는
뭔가요?
자세히 좀
알려주세요.
으악…
깊게 알려고
하고 있다!
루시! 헬프~!!

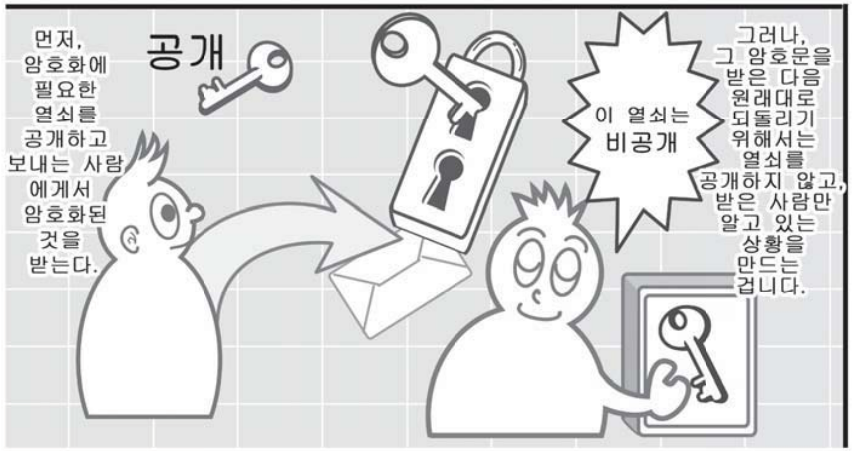

아오야마군, 작년 착실하게 학점 받았죠? 제대로 하지 않으면…
EMI가 아니고 PKI !!
죄송합니다..
원래 암호란, 특정인에게만 전달되기 위해서 그 이외의 사람이 알 수 없는 방법으로 나타내는 것입니다.


PKI를 각각 4문자씩 옮기면
HIJKLMNOPQRST
HIJKLMNOPQRST
TOM이 된다.
예전부터 사용되고 있는 암호는 문자를 조금씩 옮기는 간단한 것이 있습니다.
소위, 시저의 암호인거지. 덧붙여서 영화 '2001년 우주여행'에 등장하는 컴퓨터 HAL은IBM을 한 문자씩 옮긴게지


그랬었지, 더욱 강력한 암호가 여러가지 생각되어 왔지만..
만든방법, 푸는방법을 공유
열쇠로 비밀유지
거기에는 암호를 보내는 사람과 받는 사람이 만드는 방법, 푸는 방법을 공유해서 암호의 열쇠가 되는 것을 비밀유지하는 것이 필요했습니다.


그런거구나. 열쇠를 도둑맞으면 끝이네요.
그래. 열쇠는 제대로 관리하지 않으면 안돼. 그래도.
이용자가 많은 web 사이트에서 이용자가 열쇠를 관리하는건 힘든 일이지. 안전에 신경을 쓰고 있는지도 걱정되고


그래도 이용자에게 열쇠가 사용되지 않으면 암호라고도 할 수 없잖아요.
거기서, 이 문제는 발상을 바꿔서
암호화와 복호화 (암호를 원래로 되돌리는 것)로 나누는 것으로 해결되었습니다.


먼저, 암호화에 필요한 열쇠를 공개하고 보내는 사람에게서 암호화된 것을 받는다.
공개
이 열쇠는 비공개
그러나, 그 암호문을 받은 다음 원래대로 되돌리기 위해서는 열쇠를 공개하지 않고, 받은 사람만 알고 있는 상황을 만드는 겁니다.


그러면,
이용자에게
열쇠를
준비해서
나누어 주는것
하지 않아도
완료될
겁니다.
공개키
비밀키
이러한
암호를
공개키암호
라고
부릅니다.
흠. 흠.
그래도 열쇠가
공개되는건
편리하지만,
가짜 공개키가
간단히
위조되는거
아닌가요?
A가 아닌 사람이
A인척 한다거나…
좋은
질문이에요!
그것을
위해서…
인증국
A의 공개키
라는 것을
증명해주는
겁니다.
'인증국' 이라
불리는
국제적인
기관이
있는 겁니다.


인증국에인증되면,
인증국
< 공개키증명서 >
'공개키 OO은
A의 공개키입니다.'
라는 메세지와
인증국의 전자서명이
붙여집니다.
이것을 공개키인증서
라고 부릅니다.
전자서명에 대해서는
나중에 자세히
설명하겠습니다.
진짜 A의 열쇠인가..
인
정
!!
어디
보자~


공개키증명서를 사용하고 있는 가까운 예는
URL이 'https' 로 시작하는 Web사이트 입니다.
이거좀
빨리
조사해!
보통은 'https' 로 시작하는 Web사이트와
별다른 차이없이 그 사이트의 페이지가
표시되지만, 실제로 보이지 않는
곳에서는 공개키증명서에 대한
처리가 이뤄지고 있습니다.


'https' 로 시작하는 Web사이트에 접속하면
https
여기
두개
증명서
Web 사이트
처음에 그 사이트로부터
공개키 증명서와 공개키가
보내져 옵니다.
그렇다고해도,
공개키와 상대의
Web사이트로부터
자동적으로
보내집니다.


웹브라우저가 처리를 해주지만 사용자는 의식할 필요가 없습니다.
그렇군요.
인증국의 공개키는 이전에 브라우저에 설치되어 있기 때문이지.


웹브라우저는 먼저 공개키 증명서의 인증국이 신뢰할 수 있는 기관인지를 확인합니다.
으흐흐흐
인증국
수상해!
흠…
그 다음, 전자서명을 조사하여 공개키가 고쳐진 곳이 있는지, 공개키증명서가 유효한지, 그리고 웹사이트의 URL이 올바른지 등을 확인한다.

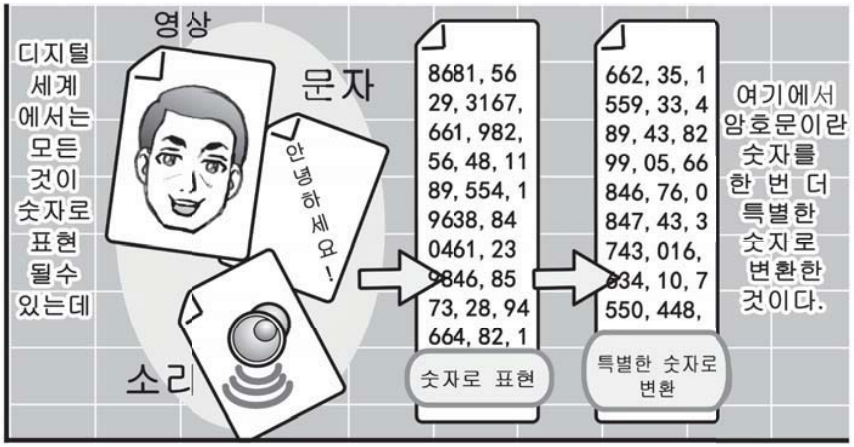

문제가 없을때는 그 웹사이트의 페이지를 보여줍니다만…
만약 문제가 있으면
(IE 7)
이러한 표시가 나옵니다.


그럼, 그러한 경고가 나오지 않으면 안심할 수 있는거에요?
NO!!
Web 브라우저에서 그러한 경고가 나오지 않도록 하는 방법도 있어.
그러니깐, 인터넷카페(PC방) 같은 곳에서 경고문이 나오지 않는다고 해서 안심하면 안됩니다.


위험하다고 생각될때는 공개키증명서의 내용을 확인하세요.
확인하다니 어떻게 하죠?

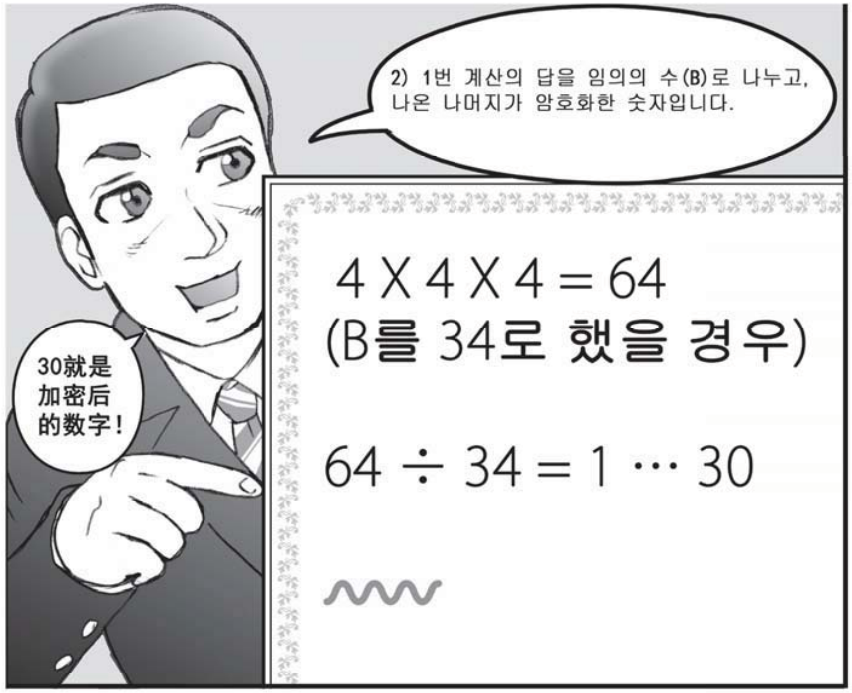

'https' 로 시작하는 web사이트의
web 페이지에는
'키마크' 라는 것이 표시됩니다.
키마크
보통, 윈도우 우하단이나
툴바등에 표시됩니다.
그 키마크를 클릭하거나
더블클릭하면,
증명서 발행기관이나
유효기관 등
공개키증명서에 대한
정보가 표시됩니다.
의심스러울때는,
'https' 라고 해도
신용하면
안됩니다.
네 !
그럼, 다음은 아까
죠 선생님이 말씀하셨던
'전자서명' 에 대해
아오야마군이
설명할 수 있겠어요?
두근
엣?
부..분명히
공개키
암호와
관계가
있는것
같은데…
Yes!
먼저,
공개키 암호의
구조를
이해해야 하죠.


공개키암호화
라는 하나의
표현을 쓰지만,
구체적인
방법은
여러가지
입니다.
실제로는
공개키암호 이외의
다른 방법을
함께 사용하고
있습니다.
공개키암호에서
현재 표준으로
사용되고 있는 것은
'RSA암호' 라고
불리는 것입니다.
앞에서, 공개키에
암호화한 암호문은
비밀키를 통해서만
원래의 글로
되돌릴 수
있다고 했는데
RSA
RSA암호에서는
그것뿐만이 아니라,
비밀키로 암호화한
암호문을
공개키를 이용하여
원래대로 되돌리는
것까지 할 수
있습니다.
아오야가
쓴거예영
비밀키
이것을 이용하면,
'이건 내가 쓴
것입니다.'
라는 것을
증명할 수 있다.
공개키
그렇구나!!


디지털 세계에서는 모든 것이 숫자로 표현될수 있는데
영상
문자
소리
안녕하세요!
8681, 56 29, 3167, 661, 982, 56, 48, 11 89, 554, 1 9638, 84 0461, 23 9846, 85 73, 28, 94 664, 82, 1
숫자로 표현
662, 35, 1 559, 33, 4 89, 43, 82 99, 05, 66 846, 76, 0 847, 43, 3 743, 016, 034, 10, 7 550, 448,
특별한 숫자로 변환
여기에서 암호문이란 숫자를 한 번 더 특별한 숫자로 변환한 것이다.


RSA암호는 다음과 같은 순서로 암호화가 됩니다.
Let's study!!

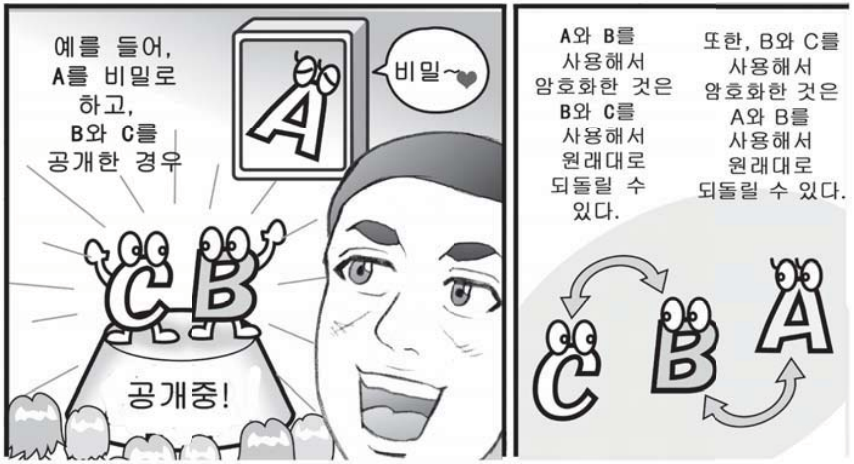

1) 먼저 기준이 되는 숫자를 임의 수(A)만큼 반복해서 곱합니다.
< 예 >
4 X 4 X 4 의 경우
3번 곱했기 때문에
A = 3


2) 1번 계산의 답을 임의의 수(B)로 나누고, 나온 나머지가 암호화한 숫자입니다.
30就是加密后的数字!
4 X 4 X 4 = 64
(B를 34로 했을 경우)
64 ÷ 34 = 1 … 30

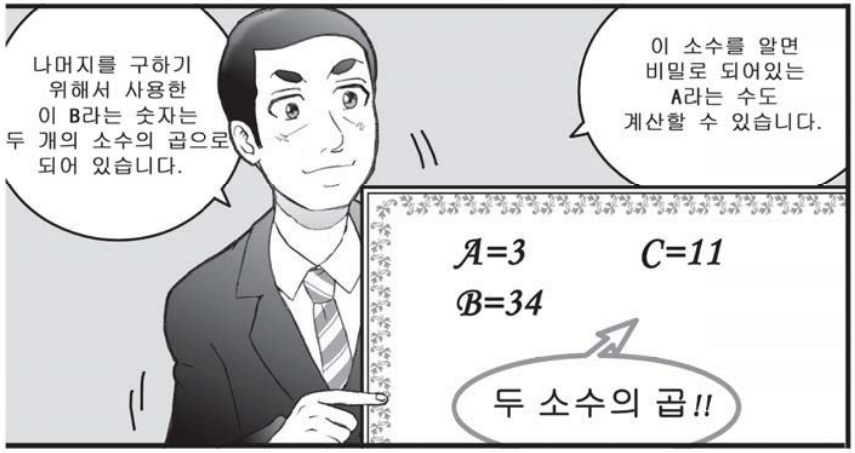

복호화해서 원래대로 되돌리는 것은 동일하게 해줍니다.
아하!
원래 숫자 4!
1) 암호화한 수를 임의의 횟수(C)만큼 반복해서 곱합니다.
2) 다음에 앞에서 계산했던 수(B)로 나누고 나머지를 구한다.
이 나머지가 원래의 숫자이다.
복호화 < 예 >
C가 11인 경우
30 X 30 X 30 X 30 X 30 X
30 X 30 X 30 X 30 X 30 X 30
30^11 ÷ 34 => 나머지 = 4

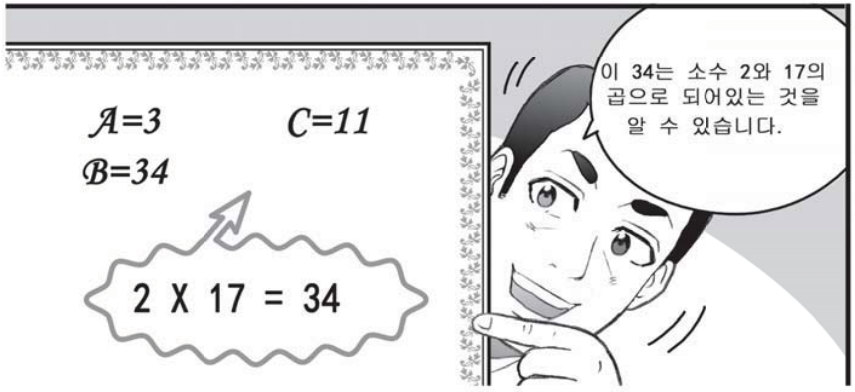

암호로 만드는 순서도
원래로 되돌리는
순서도 모두
정해져 있는 거네요.
Yes!
예전에는,
보내는 사람과
받는 사람이 암호를
만드는 방법,
푸는 방법이나,
암호의 열쇠를
비밀로 공유해야
했지만

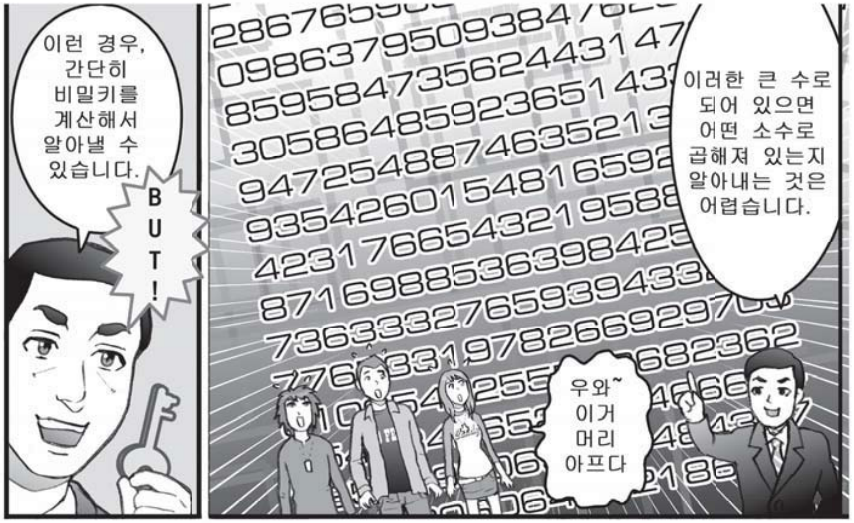

비밀키를
안전하게
공유하는게
어렵기 때문에
공개키암호라면
순서도 공유키도
공개해서,
비밀키만 안전하게
관리하면 되니까
안심이 되지요.
C B A
RSA암호에서는
A, B, C 쪽이
암호의 열쇠가
되는 겁니다.


예를 들어,
A를 비밀로
하고,
B와 C를
공개한 경우
A
비밀~
C B
공개중!
A와 B를
사용해서
암호화한 것은
B와 C를
사용해서
원래대로
되돌릴 수
있다.
또한, B와 C를
사용해서
암호화한 것은
A와 B를
사용해서
원래대로
되돌릴 수 있다.
C B A


B는
암호를
만들때도,
원래대로
되돌릴 때도
쓰는거군요.
B
맞아요.
실은 이 B가
이 암호의 강력한
점 입니다.
A, B, C의
숫자는
어떤 수를
넣어도
되는 것이
아닙니다.
RSA암호의
개발자는 어떤
수를 넣어야
수식이
성립하는지를
나타내었다.


나머지를 구하기
위해서 사용한
이 B라는 숫자는
두 개의 소수의 곱으로
되어 있습니다.
이 소수를 알면
비밀로 되어있는
A라는 수도
계산할 수 있습니다.
A=3
C=11
B=34
두 소수의 곱!!


A=3
C=11
B=34
2 X 17 = 34
이 34는 소수 2와 17의
곱으로 되어있는 것을
알 수 있습니다.


이런 경우, 간단히 비밀키를 계산해서 알아낼 수 있습니다.
BUT!
이러한 큰 수로 되어 있으면 어떤 소수로 곱해져 있는지 알아내는 것은 어렵습니다.
우와~ 이거 머리 아프다


이러한 수가 어떤 소수의 곱으로 되어 있는지를 발견하기 위해서는
알려져 있는 소수들 중에서 하나씩 하나씩 대입해 보는 수 밖에 없습니다.
세계에서 가장 빠른 컴퓨터를 사용해서 1초에 천조번 계산할 수 있어도 엄청난 시간이 걸립니다.


따라서, 현실적으로 이 공개키암호는
안 전
안전하다고 할 수 있습니다.
RSA


정부는 이 암호기술을 사용해서 전자정보등의 여러가지 서비스를 하려고 해요
그것에 기반이 바로 PKI 입니다.
각종 접수신청


그리고 PKI의 신뢰성을 보장하는게
바로 공개암호화와 소유자의 증명을하는 인증국 입니다.


그럼, 안심해도 된다는거네
현실적으로는 그렇죠.


다만 공인인증서는 본인이라는 증명을 하는 것이긴 하지만
상대가 신뢰로운 사람인지까지는 증명해 주지는 않아요
네~에!


근데 벌써 3분이 지났어!
?


내가 만든 암호 해결 못했으니까 벌칙으로 잠자고 있던 사진을 보내버리겠어~
와우
저기! 그건 절대 공개해서는 안 되요!
치사해요!

漫画：門倉つりっツ貴浩

用語集

【자료】

공개키번호

공개키번호에서는 암호화와 디코딩에 다른 키를 사용한다. 이들 키는 하나의 짝으로 되어있어 그 속의 한쪽 키를 공개한다. 메시지를 상대로부터 보내지는 경우를 생각해 본다. 상대가 암호화 할 때에 사용하는 키를 공개해 두고 그것을 사용하여 암호화되어 송부 받게 한다. 그것을 디코딩 할 때에 사용하는 키는 자신만이 가지고 있으면 메시지는 안전하게 받을 수 있다. 공개하는 키를 공개키, 비밀로 하는 열쇠를 비밀키라고 한다. 공개키번호의 방식은 몇 개나 제안되고 있지만 현재 현실에 사용되고 있는 것은 다음 항에 있는 RSA 암호다. 예를 들면 SSL에서의 (https로부터 시작되는 URL에 있어서의) 통신은 일상적으로 사용되고 있다.

RSA암호

리베스토, 샤미아, 에델먼에 의해 1976년에 개발된 공개키번호를 3명의 머리 문자를 따서 RSA암호라고 부른다. 암호화 및 디코딩은, 정수완 인수 계산을 사용한다. 해독에는 소인수분해가 필요하여 최고성능의 컴퓨터를 구사해도 소인수분해가 현실적으로 불가능한 큰 정수를 사용하는 것으로 보안 강도를 보증한다.

인증국

공개키번호를 시스템으로서 성립되게 하기 위해서는, 열쇠의 소유자가 특별히 지정되어 안전하게 관리되지 않으면 안 된다. 이것을 짊어지고 있는 것이 인증국이다. 인증국은 국제적 기관으로 정보 보안에 관해 신뢰할 수 있는 기관이 아니면 안 된다.

전자서명

적당한 문장과 그것을 비밀키로 암호화한 암호문을 보내면 수신자는 암호문을 공개키로 디코딩 하고, 그것을 암호화 전의 문장과 비교하여 일치시킴으로써 비밀키의 소유자가 보낸 것임을 확인할 수 있다. 여기에서 이것을 전자서명이라고 부른다. 실제로는 송신한 텍스트를 요약 함수를 사용하여 압축한 문장을 비밀키로 암호화해서 보내면 수신자는 보내진 텍스트를 같은 요약 함수를 사용해서 압축하고 디코딩 한 문장과 비교하는 것이 행해지고 있다.

공개키번호의 장치

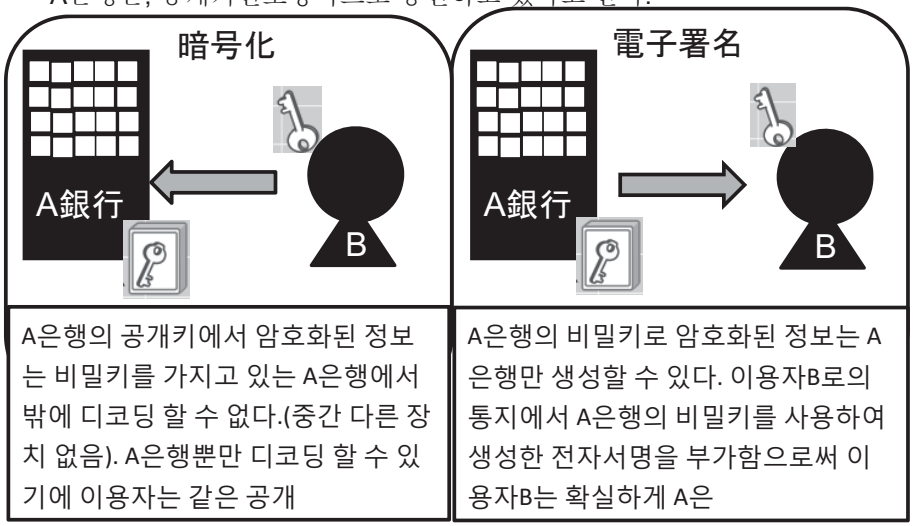
현재의 사회에서는 정보통신 속에서 암호화가 일상적으로 사용되고 있다. 암호는 왜 필요한 것일까? 예를 들면 인터넷쇼핑이나 온라인 은행에 있어서의 정보의 교환에서는 금전의 수수가 행하여진다. 그 때 정보가 외부에 새는 것은 용서되지 않는다. 인터넷을 할 때에는 내용을 제 삼자에게 알리지 않고 상대에게만 확실하게 정보를 전하는 것 (다른 사람에게는 흘리지 않는다)이나, 확실하게 발신자를 특정시킬 수 있어 만약 내용의 개편이 발생하면 그것을 검출 시킬 수 있다. (개편, 위장을 막는 것) 이 필요하다. 여기에서 전자를 위해 암호화가, 후자를 위해서 전자서명이 사용되고 있다.

웹 사이트상의 정보 교환으로 인터넷이 사용된다. 인터넷에서는 정보의 분리(패킷화)로 하여 전하며, 통신 경로에는 각양각색의 중계 기기(라우터)가 존재하고 있다. 통신 내용을 암호화함으로써 도청을 막을 필요가 있다. 또 암호화된 정보의 디코딩은 대단히 어려운 것이 아니면 안 된다.

일반적으로 기업은 다수의 고객을 안고 있다. 암호를 위해서 고객고유의 키를 매회 생성, 배포하는 것은 곤란하다. 이것을 해결하는 것이 공개키번호 방식이다. 공개키번호에서는, 이하의 성질을 가진 공개키와 비밀키의 두 가지 키를 사용한다.

■공개키로 암호화한 메시지는, 비밀키에서 밖에 디코딩 할 수 없다.
■비밀키로 암호화한 메시지는, 공개키로밖에 디코딩 할 수 없다.

A은행은, 공개키번호방식으로 통신하고 있다고 한다.



A은행의 공개키에서 암호화된 정보는 비밀키를 가지고 있는 A은행에서 밖에 디코딩 할 수 없다.(중간 다른 장치 없음). A은행뿐만 디코딩 할 수 있기에 이용자는 같은 공개

A은행의 비밀키로 암호화된 정보는 A은행만 생성할 수 있다. 이용자B로의 통지에서 A은행의 비밀키를 사용하여 생성한 전자서명을 부가함으로써 이용자B는 확실하게 A은

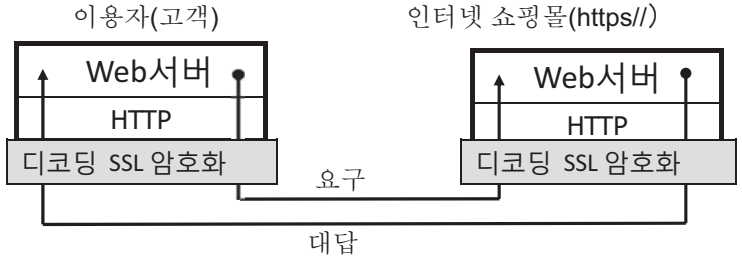
실제의 SSL통신

공개키번호가 제안되기 전에는 암호라고 하면 공통의 열쇠를 이용하여 암호화· 디코딩을 하는 것이었다. (공통키암호). 공통키에서의 큰 문제는, 키를 어떻게 안전하게 보내는 사람과 수신자가 공유할 지다. 인터넷상에서의 교환에서는 열쇠를 직접 상대에게 주고받을 수 없다. 안전하게 열쇠의 수수를 하는 것은 어렵다. 최근에는 공개키번호의 구조가 만들어져 안전하게 메시지의 교환을 할 수 있게 되었지만 공개키번호에서는 공통키암호에 비해 처리 속도가 늦은 것이 난점이다. 이 장점·단점을 비추어 실제로는 다음과 같은 것이 행해지고 있다.

•메시지는 공통키방식으로 암호화하여 통신한다.
•그 공통키는 공개키방식으로 암호화하여 통신한다.

즉, 공개키방식으로 안전하게 공통키의 수수를 하고, 그 후는 그 공통키로 메시지를 교환하는 것이다. 그 예로서, 여기에서는 인터넷상의 쇼핑몰에 접속하여 쇼핑을 하는 것을 예로 들 수 있다.

⓪서버 상에 증명서등록완료의 공개키, 비밀키가 등록되어있다.
①이용자에게서 요구가 있으면, 서버는 공개키를 증명서부착으로 이용자에게 보낸다.
②이용자 측은 공개키가 진짜인 것을 확인하고, 공통키를 발행하여 서버에 보낸다.
③서버상에서 비밀키를 사용하여 공통키를 얻은 후 그 공통키를 이용하여 암호화 통신을 시작한다.



이러한 순서를 거침으로서 암호화된 내용을, 통신 도중에는 다른 곳에 흘리지 않고 보낼 수 있다.

RSA암호에 있어서의 문자열의 암호화•디코딩【발전항목】

여기서는, RSA암호로 어떻게 메시지를 암호화·디코딩 하고 있는지에 대한 개략을 나타낸다.

●RSA암호의 알고리즘

【암호화】

①메시지의 문자열을 정수화시킨다.
②각각의 정수에서 대하여,
암호의 정수= mod (원의 정수^{C}, B)
여기서, C, B는 공개키 (원의 정수< B)
상기의 식으로부터, 암호화된 정수를 추구한다.
한편, mod(x, y)이라고 한 표기는 x 을 y 로 나눈 나머지를 나타낸다.

【디코딩】

①각각이 암호화되고 있는 정수에서 대하여,
원래의 정수= mod(암호의 정수^{A}, B)
여기서 A는 비밀키
상기의 식으로부터, 원래의 정수를 추구한다.
②추구된 정수를 문자열화시킨다..

●공개키로 이용하는 열쇠A, B, C의 생성 방법
① 다른 소수p 및 q를 적당히 선택한다. B = p × q
② j 은, 최소 공배수(p-1, q-1)로 한다.
③ j 과 서로 근본이 되는 C를 적당히 선택한다. (최대공약수(j ,C)=1)
④ mod(C × A, j)=1 되는 A를 요구한다.

여기에서, B와 C가 공개키로 A가 비밀키이다. 해독되지 않기 위해, 큰 숫자를 사용한다. B는 2개의 소수의 적분이 되고 있다. 즉, B를 2개의 소수의 적분(소인수분해)할 수 있으면, RSA암호의 알고리즘은 위에서 말한 바와 같이 미리 결정되어 있으므로 비밀키를 계산할 수 있게 된다. 따라서 이 암호의 강도는 B가 얼마나 큰 숫자인지에 의존하게 된다. 소인수분해는 대상의 정수가 크면 클수록 급격하게 어려워진다. 현재 안전성이 높은 암호가 되기 위해서는 B가 2048비트 (2048자릿수의 0/1의 나열)이상으로 요구되고 있다. 이것은, 미국 국립표준기술연구소가 낸 정부기관에서의 사용에 필요로 되는 요건이며, 2010년까지의 이행을 요구하고 있는 것으로, 암호의 2010년 문제라고 불리고 있다.