Bogenberger - Damsell, O., Cleary E., Crowley, T., Kocik, D., McMorrow, R. (2020) Report on Cryptography, University of Limerick, unpublished

GitHub: https://github.com/QrowLee/Cs4182project

Poster: https://github.com/QrowLee/Cs4182project/blob/master/Group10poster.pdf

Abstract:

This is a report on Cryptography. We discussed the history of cryptography, different types of cryptography and their impact on Computer Science.

Contents

Introduction

Cryptography and encryption have always been an interest of people. Humans like to feel safe and that includes what they own. People want what is theirs to stay theirs and they don't want anyone to be able to steal it. From the Romans to modern day and into the future, cryptography has existed in one form or another. The introduction of Computers in recent years has made this topic much more interesting. We always need to keep thinking of new ways to encrypt and decrypt our data.

Early History

The Caesar Cypher was one of the first actual uses of cryptography to hide messages.

"if there was occasion for secrecy, he wrote in cyphers. The way to decipher those epistles was to substitute the fourth for the first letter."

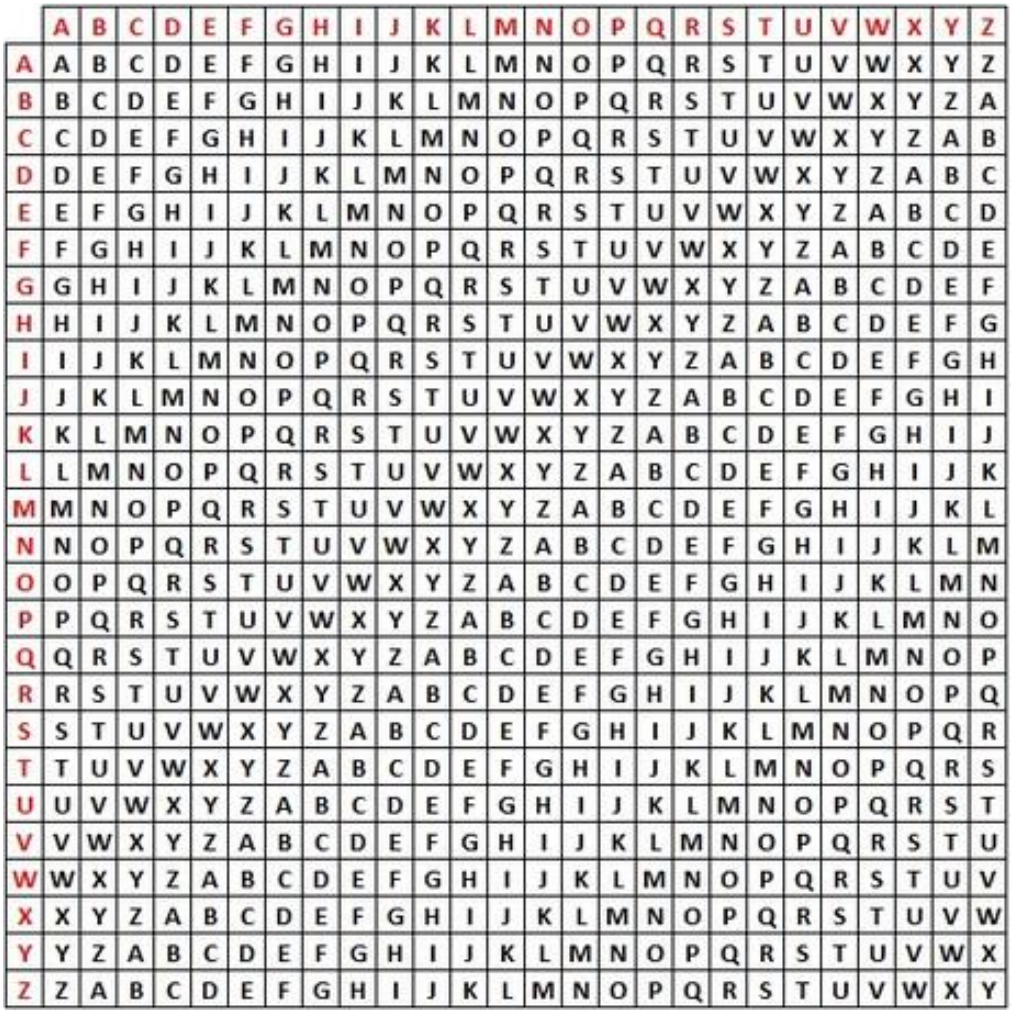
The Twelve Caesars 56.Gaius Suetonius Tranquillus

The Caesar cypher is a simple form of cryptography but as it was close to the first use of cryptography it didn't need to be complex. The Caesar Cypher was a form of substitution cypher and is even used today in the ROT13 method.

The Vignere cipher was similar to the Caesar cypher but was also the first use of an encryption key. Vignere Cypher was a Cypher first described in the book *La cifra del sig*, by Giovan Battista Bellaso in 1553 and then later misattributed to Blaise de Vigenère. The

cypher was known to be extremely secure in the era of pen and paper cryptography and so earned the name "le chiffre indéchiffrable".

The Cypher worked using a square grid of alphabets, as shown here.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Keyword was the encryption key. The encoder would write their message in plain text, and then repeat the keyword inline underneath until the lengths were equal.

thisisademonstration
pizzapizzapizzapizza

The encoder would then check on the grid where the letter in the plaintext and in the keyword overlapped in the grid, then write down that letter in the encoded message. For example, letter b in plaintext and letter p in the keyword would make q in the encoded message.

And the example shown above becomes.

IPHRIHICDMDVRSRPBHNN

Cryptography and early computing

The Enigma machine was another method of substitution encoding similar to that of the Caesar cypher. However, using a machine, a much more complex system could be developed from the same principles. The enigma machine had keys like a typewriter and a set of rotors

which changed the letters being printed by the machine, these rotors moved after every keypress of the machine. This meant that the machines would have a different setting after each keypress making the message even more difficult to decode, if an identical machine was set to the same setting however the recipient would only need to type the received message into their own machine, the machine could also have different settings by swapping out rotors and using the plugboard and so had 158,962,555,217,826,360,000 possible settings.

The Bombe machine used a method that could now be called exhaustive search to decode the setting of the enigma machine. Turing used the fact that certain words could be expected to appear in messages, e.g. heil Hitler, and that no letter could be transformed into itself to shorten the search the rotors then turn, trying combinations, until they form an open circuit telling the team the decrypted message.

Cryptography and Modern computing.

Cryptography was really only used by banks and the military until the internet was created. It was then discovered that there was a need for accessible cryptography and so the Data.Encryption.Standard(DES) was born. The DES is a form of 64-bit Block Cypher, set as a US federal Standard in 1976 and published in 1977. (U.S.A., National Institute of Standards and Technology 2012).

Secret key cryptography

Introduction

Symmetric (secret key) encryption is “an algorithmic tool that allows a pair of parties to communicate secret information over open communication media that are accessible to eavesdroppers.” (Theory of Cryptography Conference Corporate 2010) This is a classic model of encryption, where the both parties share a secret key. The key is assumed to be random, single-purpose and not-dependant on the message. The security is ensured by the fact that adversary cannot intercept the key.

According to Buchanan (2017, p.55), there are two main types of symmetric (a.k.a. “secret key”) encryption: stream cipher and block cipher. Buchanan also mentions that symmetric key decryption is faster than the asymmetric one with two keys, and so it is more suitable to use where data has to be transmitted in real-time e.g. secure voice communication online.

Stream cipher

Stream encryption works by operating on a continuous data stream, where “the message is broken into successive bits or characters and then the string of characters is encrypted using a key stream” (Nandi *et al.* 1994). In order to create ciphertext, a pseudorandom, pseudo-infinite key in binary form must be generated from initialisation vector, which acts as a random seed. This key is then XOR-ed with the plaintext (here the data stream) bits. (Buchanan 2017, p.58)

Traditionally, modulo-2 addition performed by the XOR gates was used for stream encryption. According to Paar (2010, p.33), this is because after XOR operation the resulting

bit has 50% chances of being 1 and 50% chances of being 0, unlike if we were to use other gates.

RC4

RC4 is a popular stream cipher algorithm. It is used by SSL (SecureSocketLayer) and WEP (WirelessEncryptionProtocol). RC4 is made up of key scheduling (KSA) and pseudo-random number generation (PRGA) algorithms (Isobe *et al.* 2014, pp.1-2). The first algorithm creates *S*, which is an array of 256 bytes. Element *i* of *S* is swapped with element *j* of *S*, where *j* is a sum of its previous value, value of *S[i]* and the *i*th byte of the key. Since our key is shorter than 256 bytes, we use modulo operation and traverse through the same key again. After this is completed, we run the PRGA which traverses through the vector *S* as *i* is incremented by 1 and *j* is increased by the *i*th byte of *S*. The *i*th and *j*th bytes are then swapped and the remainder of their sum divided by 256 is the index of *S* that points to the final keystream byte. As we keep getting the mod 256 of *i* and *j* as those values increase, we get pseudo-infinite key. The key is then X-ORed with our plaintext data bit-by-bit, resulting in a ciphertext.

This method, however, although very fast, is not considered completely safe. RC4 is subject to "plaintext recovery attacks" that work for the "initial bytes of the keystream", which are actively used by SSL/TLS (Isobe *et al.* 2014, p.15).

Block cipher

Block ciphers work by splitting the data into blocks. Although most blocks are fixed size, the last block could be shorter, since data length does not have to be a multiple of block size. For this reason, block encryption requires padding, which would fill the space in the last block.

Popular ways of padding include adding NULL characters, or 0x80 and then NULL characters (Bits), or "the same value as the number of the padding bytes" (CMS standard), filling with zeros until the last 8 bits, which are set to length of the added bytes ("ZeroLength") (Buchanan 2017, p.59).

There are many block ciphers in use today, such as Blowfish, AES and RC5.

AES/Rijndael

AES (Advanced encryption Standard), is a popular implementation of the block cipher, also used in WPA2. The block size is 16 bytes and the key used for encryption can have 128, 192 or 256 bits (A.E.S. Corporate 2005). To easily represent AES block, split it into bytes and form a square matrix of length 4. The operations performed on a block consist of 10-14 rounds (depending on key length). Each round performs operations "AddRoundKey", "SubBytes", "ShiftRows" and "MixColumns". The columns are not mixed during the final round (A.E.S. Corporate 2005, p.2).

AES types include Galois/Counter Mode that XORs each block with the next, which adds on the mechanics from the stream cipher, and Cipher Block Chaining, where the initialisation vector is used for encryption of the first block and the vector is sent to receiver who decrypts that block. Each consecutive block is XORed with the previous block to decrypt the sequence (Buchanan 2017, pp.65-66).

Public key cryptography

Introduction

Public key encryption or asymmetric key encryption uses both a public and private key to secure communications between two entities. The public key is distributed while the private key is kept private. The mathematics of the encryption make it difficult to determine one key when given the other because of the difficulty in factorizing a value for its prime number factors. (Buchanan, William J. Cryptography, pg. 144) This is achieved using the following methods.

Integer factorization (RSA method) **discrete logarithms** (ElGamal) and **elliptic curve relationships** (Elliptic Curve).

The public key checks the identity of the entity by using its public key to decrypt a message that was encrypted using the entities private key. Since the two keys are mathematically linked then they can only be decrypted using the other which proves its identity.

It is also used for the protection of a symmetric key. This is usually used in disc encryption where the symmetric key that is used to encrypt a file is protected with the public key of an entity so the only key with access to the symmetric key is the private key.

It's most common use is to establish secure communications between two entities by exchanging their public keys and using them to encrypt the data they want to send to. That way the only entity that can decrypt the message is its intended recipient.

The key is usually stored in an XML format or on a digital certificate which allows it to be stored processed and transmitted this is known as the (PKI) or the Public Key Infrastructure. Key pairs are generated by trusted entities and the public key is distributed using the PKI. The most important part of the PKI is the keeping of the private key secret, because if it loses its secrecy then the entities security and identity could be breached along with the any encryption keys that are protected by the key pair.

RSA

The public and private keys are generated from very large prime numbers as a value which is the product of another two large prime numbers that is extremely difficult to factorize. The public key is then passed where it is used to encrypt data intended to be sent to the entity.

The most used and well known algorithm in RSA selects two large prime numbers that are usually around 256 bits in length a and b and are also to the order of 10^{100}, these factors are then kept secret and you make the modulus N by multiplying them together. Then to make the second public key you choose another value e so that e and $(a - 1) \times (b - 1)$ are relatively prime as in they do not share a common factor bigger than 1 so GCD $(x, y) = 1$. The public key is then $<d, N>$ and this makes a key that is at least 512 bits long.

The private key for decryption d is computed so that:

$$d = e - 1 mod[(a-1) \times (b-1)]$$

Or

$$(d \times e) mod[(a-1)(q-1)] = 1$$

PHI is defined as $(p-1) \times (b-1)$. The encryption process to the *c* is = m^e *mod N,* and the message *m* is then decrypted with the formula *m* = c^d *mod N.*

EIGamal

The keys are generated using a description of a cyclic group *G* of order *q* with the generator *g*. *e* is represents the unit element of *G*. A random integer *x* is then chosen from

$\{1, \ldots, q-1\}$ $h := gx$. The public key is made from the values *(G, q, g, h)* and *x* is kept as the private key.

The message is encrypted by mapping it to an element *m* of *G* with a reversible mapping function. You choose another random integer *y* from $\{1, \ldots, q-1\}$. Compute the shared secret $s := hy$ as well as $c1 := gy$ and $c2 := hy$ $c2 : m \times s$ and send the cipher text $(c1, c2)$ to the message recipient.

$(c1, c2)$ is decrypted with the private key *x* by computing $s := c1x$ because

$$c1 := gy, c1x = gxy = hy$$

Which is the same shared secret used. s^{-1}, the inverse of *s* is then computed in the group *G*. You then compute s^{-1} as c_1^{q-x} because of Laranges theorem which states

$$s \times c = gxy \times g(q-x)y = (gq)y = ey = e.$$

$m := c2 \times s - 1$ is then computed producing the original message because

$$c2 = m \times s \; so \; c2 \times s - 1 = (m \times s)s - 1 = m \times e = m$$

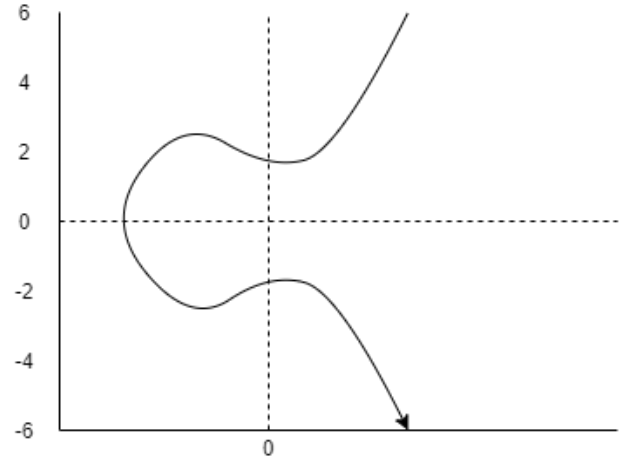
The result *m* is then mapped onto the message.

Elliptic curve

Elliptic Curve ciphers (ECC) are better than RSA for use in embedded systems because of the high overhead RSA has on processor loading along with the power drain and requirements for the memory. The main advantages are that it has much smaller keys where the prime number P is normally only 160 bits, making it allot faster to encrypt. The generation of the curves is more difficult so it's harder to crack, they can also be used to factories values e.g. finding the prime numbers in RSA.

The curve takes the form $y2 = x3 + ax + b$ and the equation for the curve is

$$y2 = x3 + ax + b(mod\ p)$$



x, y, a and b are within Fp and are all integers. *a* and *b* are coefficients of the curve where the curve fulfills the condition:

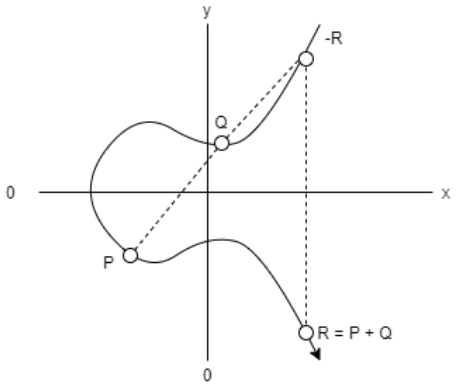
$4a3 + 27b2 \neq 0$. This makes sure there are no singularities in the curve. The curve

is horizontally symmetrical and a non-vertical line intersects the curve at 3 points.

If you select two points *P* and Q on the curve and draw a straight line between them we get *nP=Q* where *n* is a scaler. When you are given *P* and *Q* you can't find *n* if it is large enough.

We then find the value of *n* given *P* is very large and n is also large it is easy to find *nP*, but when only given *P* and *nP*, it is almost impossible to find *n.* we can find *P* if we have *2P* (*n=2*), however it is extremely difficult to find *n* when it is so large. "*n* is the discrete logarithm between *P* and *nP*, and the main operation is multiplication, this is different to prime number factorization in RSA". (Buchanan 2017, p.156)

$$y^2 = x3 + ax + b(mod\ p)$$



Hash functions

Introduction

A hash function is a function that can be used to map data of any size to a number of fixed size. The values returned by a hash function are called hash values, hash codes, digests, or hashes. The values are used to index a fixed-size table called a hash table. Using hash functions to index a hash table is called hashing (Knuth 1973, p.527).

Hashes do not need to be able to be decrypted. If 2 pieces of data have the same hash, then it is (probably) the same file.

A hash function takes an input as a key, which is associated with a method and used to map it to the data storage and retrieval application. The keys may be fixed length, like an integer, or variable length, like a name. In some cases, the key is the data itself. The output is a hash code used to index a hash table holding the data or records, or pointers to them.

A hash function has three requirements:

1. It must be fast, but it can't be too fast or else it becomes too easy to crack.

2. It must have an avalanche effect. Changing one bit ANYWHERE in the file should change the entire Hash.

3. It must be able to avoid collisions. Think of an analogy with cables. If there is a collision with Hashes, then it's like trying to plug 2 cables into the 1 socket.

MD5
A Hash function is considered broken if it's possible to create collisions deliberately. MD5 was previously the most widely used but now it is considered broken for this exact reason. People found out how to deliberately create collisions.

The problem with this is that if you are able to intercept a file and edit it and have the hash stay the exact same then you can put something malicious into the file and cause problems.

MD5 is so broken now that you can figure out what its hashes represent by typing them into google. People used to store passwords this way and if anyone were to use MD5 for a password nowadays someone could crack it by just looking it up on Google.

SHA1, 2 and 3
People then began to move to SHA-1 which was created by the NSA. However now it's thought that this might start to become broken as well as computers get faster and faster.

Some moved to SHA-2, which for the time being is secure.

SHA-3 is being checked by agencies. In a few years this will become the standard.

None of these should ever be used for storing passwords. They can become broken too easily. They should only be used for file transfer.

Hash tables
Hash functions are used with Hash table to store and retrieve data items or data records. The hash function translates the key associated with each piece of data into a hash code which is used to index the hash table. When an item is added to the table, the hash code might index an empty slot (bucket), in which case the item is added to the table there. If the hash code indexes a full slot, some kind of collision resolution is required (Mendezes *et al.* 1996).

The new item may:

- not be added to the table
- replace the old item
- added to the table according to some other rules.

Crypto cracking
Introduction
Encrypted data often contains private information, and adversaries might try to crack it. Data is only secure if the key stays secret. While one method of decrypting a message might not work, use of multiple techniques and deduction could result in the cipher being broken.

Brute force attacks
Brute force attack is when an attacker tries every possible key combination. This process is usually automated, and it relies on checking different permutations of values for the key.

The key entropy is important here since it "measures the amount of unpredictability, and in encryption it relates to the degree of uncertainty of the encryption process" (Buchanan 2017,

pp.82-83). Keys are often formed from dictionary pass phrases, so certain character combinations are more probable than others. Entropy is given by logarithm base 2 of N, where N is the number of possible permutations of ASCII characters to form a phrase of that length. The higher the entropy, the more unpredictable the key and harder it is to crack the cipher.

Cryptoanalysis

According to Knudsen and Mathiassen (2000, p.2), cryptoanalysis is an attempt to find a relation between the key, plaintext and ciphertext. A cracker could know a specific pattern in plaintext, and then could locate it in the ciphertext. This would allow them to find the key used for encryption and decrypt the rest of the message. This is known as "known plaintext attack".

Another method could be "chosen ciphertext". An adversary could create their own message and send it to a server. This server would then receive, encrypt the message with its private key. The ciphertext formed could be intercepted, analysed and compared with the original message to find the key used by the server.

Man-in-the-middle attacks occur, when an intruder pretends to be the intended message recipient. The adversary goes unnoticed as communication seems to occur as normal, except the man-in-the-middle would be in full control of data.

The intruder could also make a blinding attack, which, according to Buchanan (2017), consists of making a message in form $M'=r^{**}e\ M(mod\ N)$, where r is a random number, M is the message we will want to send, and e is the exponent of the user's encryption key. This message is then sent to be signed for and then from that by diving by r we can get a working signature for our message M. Then, we could send a signed message to a server. In real life, the message sent to the other party could be a different account number sent to a bank's server to pay our money to.

Use of backdoors

Some cryptographic algorithms initially had secret backdoors that would allow the government agencies to break the encryption. The keys used by the citizens could be registered and kept in an escrow (Buchanan 2017, pp.250-252). Another backdoor would be a NOBUS, which stands for "NObody BUt US", which means that there is a mathematical way of breaking the cipher that only authorities know about.

VPN and blockchain in cryptography

VPN

A VPN, or Virtual Private Network, allows you to create a secure connection to another network over the Internet (Cantrell *et al.* 2006). VPNs can be used to access region-restricted websites, shield your browsing activity from prying eyes on public Wi-Fi, and more.

In a VPN, Cryptography has a big part to play since the Virtual Private Network takes a lot of encryption (the process used to convert information or data into code, mainly to stop unauthorised access). The encryption is necessary for the data protection of the network.

Without this, the VPN wouldn't have cryptography in it or even have a decent security system. The features that are important for a good VPN are:

1. Security
2. Reliability
3. Scalability
4. Network management
5. Policy management

There are two main types of VPN:

- Remote access: Also referred to as a Virtual Private Dial-up Network (VPDN). These are a user-to-LAN connection, mainly used by companies who have a lot of staff working from remote areas that need to connect to the private network. When a company wants to set up large remote-access, VPN provides a form of internet dial-up account to their employees using an Internet Service Provider (ISP).
- Site-to-site: Some companies can connect multiple fixed sites over a public network such as the internet. To pull this off, the company needs advanced equipment and a lot of large-scale encryption. This is very beneficial since if it is used right, each site will only need a local connection to the same public network, thus saving money on long private lease lines.

Blockchain

In the simplest terms, Blockchain can be described as a data structure that holds transactional records and while ensuring security, transparency, and decentralization.

In the blockchain, digital encryption technology has a core position. The security of user information and transaction data is a necessary condition for the promotion of blockchain. The development of cryptography technology promotes and restricts the further development of blockchain. The most successful blockchain that cryptography effected has to be Bitcoin, the online currency.

Blocks in a blockchain

"Blocks" on the blockchain hold pieces of digital information, this is where most of the cryptography takes place, specifically they have three parts:

1. Blocks store information about transactions like the date time and euro amount of your most recent purchase from an online site of your choosing.
2. Blocks store information about who is included in the transactions. for example, if you bought a new laptop off done deal, a block would record your name along with donedeal.ie. Instead of using your real name, the block records your purchase without any identifying information using a unique "digital signature" sort of username.
3. Blocks store information that distinguishes them from other blocks.

Bitcoin

We cannot talk about the blockchain without coming to mention Bitcoin, Bitcoin uses a lot of cryptography and has even made its own currency worthful against other types of currencies such as euro's, dollar's and yin. When bitcoin was created, cryptography wasn't commonly used and thus, its value was nil, but as time went on and cryptography began to grow in the

world, the value of the currency began to rise, you used to be able to buy a lot of bitcoins for cents, now as of today's date, 11/05/2020, a singular bitcoin is worth about 8,725.47 dollars.

Future of cryptography with quantum computing

Quantum computing

Computers are constantly getting faster as we develop new technology to do tasks in fractions of the time. This means we need to make the parts smaller and smaller so we can fit more power into the components. Computer parts are starting to approach the size of an atom. At this scale physics as we know it breaks down and a new rule set appears. Transistors block the passage of electrons, this is the basis of all computing. At these incredibly small scales however electrons can just pass through the transistors using a process called quantum tunnelling. The next step in computing which people are currently working on is Quantum Computing. Quantum computers use Qbits rather than bits. Qbits can be set to the values of 0, 1 or a state in-between. It can be either but once you measure the Qbit it has to pick 0 or 1. Let's say you have 4 bits and 4 Qbits. The 4 bits can only be in 1 of the 16 possible combinations at any given time. Qbits however can be in every single one of those 16 combinations at the same time.

Impact on cryptography

A Quantum Computer will be superior to a regular computer when it comes to dealing with large amounts of data. Certain things like searching through data or checking possible combinations become faster with a Quantum Computer. Quantum Computers only require the square root of the time a normal computer would. In banking, you give people a public key which is used to encode messages only decodable using your private key. The public key can be calculated using the private key but with regular computers this would take too long. It would be trial and error until it figures all the maths out. Quantum Computers could do the maths for it much faster. This makes current cryptography almost useless. People will need to start working on new ways of security and encryption. Maybe in the future people will use Quantum Computers to break security created by another Quantum Computer. Fighting fire with fire if you will. Right now, we aren't anywhere near where Quantum Computers will be widely used. They will probably never replace the household PC but their uses in society are definitely going to be helpful in progressing our knowledge about the Universe.

Conclusion

Cryptography is a very large subject. It has many uses from sending secure messages to banking and hiding one's identity. We can never for sure keep our data safe. All we can do is try our best to make it as difficult as possible for our data to fall into the wrong hands. No code is unbreakable. If some encryption is impossible to decrypt, then it's useful to nobody. That is just the nature of Cryptography.

List of references

A.E.S. Corporate Author (2005) 'Advanced Encryption Standard - AES 4th International Conference', AES 2004, Bonn, Germany, May 10-12, 2004, Revised Selected and Invited Papers', available: http://dx.doi.org/10.1007/b137765.

csrc.nist.gov

Buchanan, W. J. (2017) *Cryptography*, Gistrup, Denmark: River Publishers.

Cantrell, C., Henmi, A., Lucas, M., Singh, A. (2006) Firewall policies and VPN configurations, Rockland, Massachusetts: Elsevier Science & Technology Books.

Crypto Corner (2013) *Tabula Recta* [image], available: https://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/1889186_orig.jpg [accessed 09 Apr 2020].

ElGamal T. (1985) 'A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms' in Blakley G.R., Chaum D., eds, *Advances in Cryptology*, Santa Barbara, United States, 19-22 Aug, Berlin: Springer-Verlag Berlin Heidelberg, 10-17, available: https://doi.org/10.1007/3-540-39568-7_2 .

Isobe, T., Ohigashi, T., Watanabe, Y. and Morii, M. (2013) 'Full Plaintext Recovery Attack on Broadcast RC4' in Moriai S., ed., *International Workshop on Fast Software Encryption*, Singapore, 11-13 Mar, Heidelberg, Berlin: Springer-Verlag Berlin Heidelberg, 1-16, available: https://doi.org/10.1007/978-3-662-43933-3_10 .

Katsikas, S., Mauw, S., Mjølsnes, Stig F. (2008) 'Public Key Infrastructure 5th European PKI Workshop: Theory and Practice', *EuroPKI 2008*, Trondheim, Norway, 16-17 Jun, Berlin, Heidelberg: Springer Berlin Heidelberg.

Knudsen, L.R. and Mathiassen, J.E. (2000) 'A chosen-plaintext linear attack on DES', in Schneier B., ed., *International Workshop on Fast Software Encryption*, New York, United States, 10-12 Apr, Heidelberg, Berlin: Springer-Verlag Berlin Heidelberg, available: https://doi.org/10.1007/3-540-44706-7_18 .

Knuth, D. 1998, *The Art of Computer Programming, Vol. 3, Sorting and Searching*, 2nd ed., Reading, Massachusetts: Addison-Wesley Professional, p. 527.

Menezes, A.J., van Oorschot, P.C., Vanstone, S.A. (1996) *Handbook of Applied Cryptography*, Massachusetts: CRC Press.

Nandi, S., Kar, B.K. and Pal Chaudhuri, P. (1994) 'Theory and applications of cellular automata in cryptography', *IEEE Transactions on Computers*, 43(12), 1346-1357, available: http://dx.doi.org/10.1109/12.338094.

Paar, C. (2010) *Understanding Cryptography: A Textbook for Students and Practitioners*.

Schmeh, K. (2006) *Cryptography and Public Key Infrastructure on the Internet*, Bochum, Germany: Wiley.

Theory of Cryptography Conference Corporate, (2010) *Theory of Cryptography 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010, Proceedings*, Berlin, Heidelberg: Springer Berlin Heidelberg.

U.S.A., National Institute of Standards and Technology (2012) *SP 800-67 Rev. 1*, available: https://csrc.nist.gov/publications/detail/sp/800-67/rev-1/archive/2012-01-23 [accessed 11 Apr 2020].

Bibliography

Cisco (2008) *How virtual private networks work*, available: https://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/14106-how-vpn-works.html [accessed 09 Apr 2020].

Computerphile (2013) 'Hashing Algorithms and Security – Computerphile' [video], available: https://youtu.be/b4b8ktEV4Bg [accessed 09 Apr 2020].

Computerphile (2017) 'SHA: Secure Hashing Algorithm - Computerphile' [video], available: https://www.youtube.com/watch?v=DMtFhACPnTY [accessed 09 Apr 2020].

HowStuffWorks (2019) *How a VPN (Virtual Private Network) works*, available: https://computer.howstuffworks.com/vpn7.htm [accessed 09 Apr 2020]

Investopedia (2020) *Blockchain explained*, available: https://www.investopedia.com/terms/b/blockchain.asp [accessed 10 Apr 2020].

Kurzgesagt – In a Nutshell (2015) 'Quantum Computers Explained – Limits of Human Technology', *Futurism* [video], available: https://www.youtube.com/watch?v=JhHMJCUmq28 [accessed 10 Apr 2020].