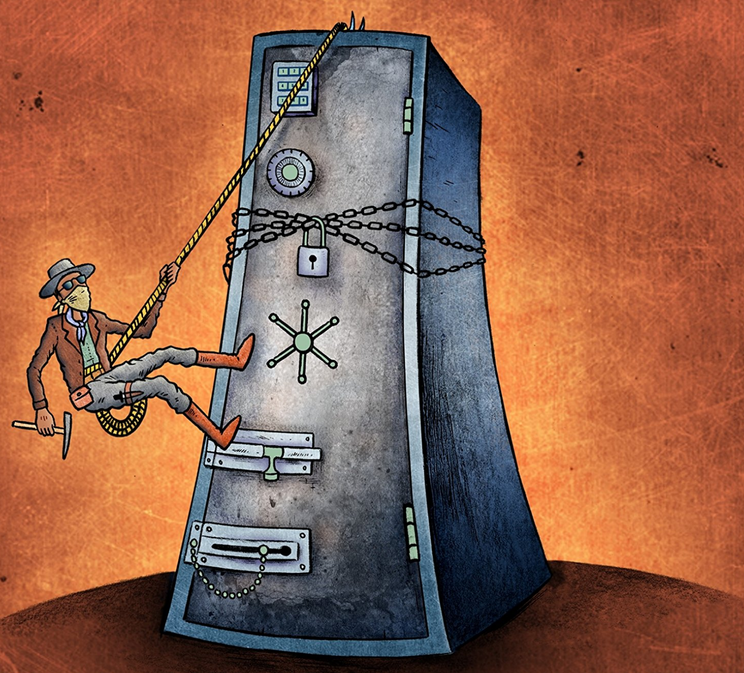
ENCRYPTION WITHOUT MAGIC,

RISK MANAGEMENT WITHOUT PAIN

#qconlondon @vixentael

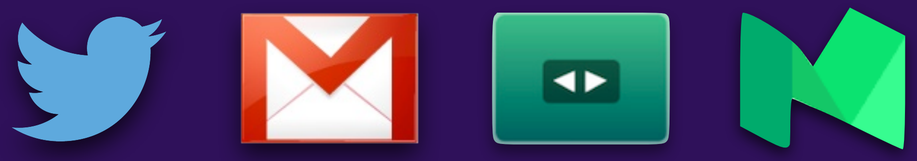
Serious Cryptography



Jean-Philippe Aumasson

no starch press

@vixentael



Feel free to reach me with security questions.

I do check my inbox :)

Product Engineer

COSSACK LABS

CRYPTOGRAPHY?

Blowfish

Twofish

Rabbit

Salsa20

AES

OFB

SEED

CFB

CRYPTOGRAPHY?

RSA

DSS

ECDSA

DES

Camelia

SEAL

CBC

DSA

3DES

SHARK

RC4

ECB

CTR

#qconlondon @vixentael

~~MD5~~

~~SHA1~~

SHA3

CRYPTOGRAPHY

key management

algorithms

public key validity

elliptic curves

storing secrets

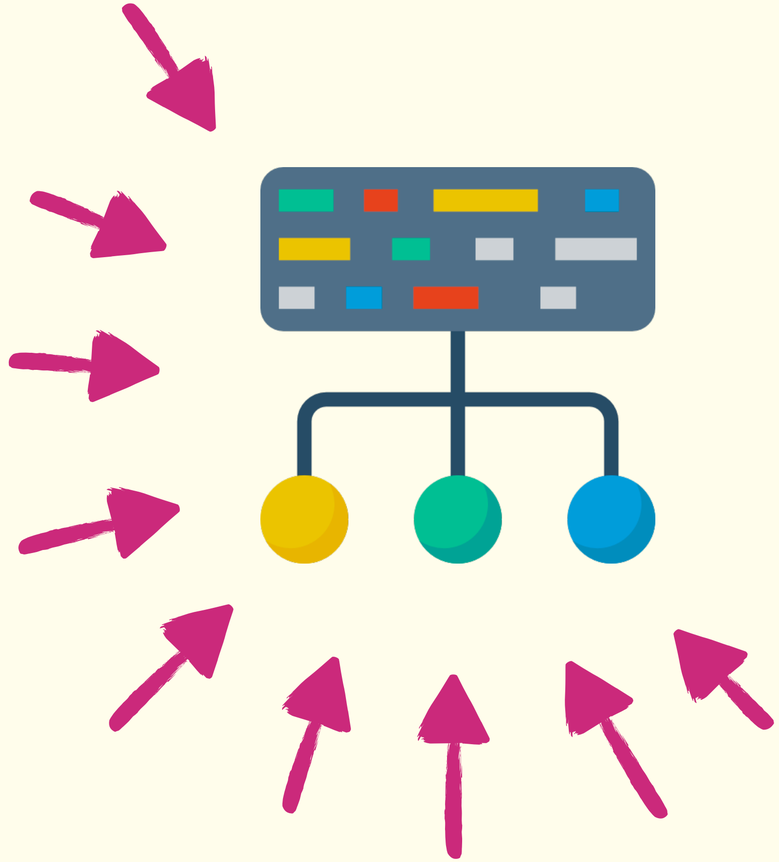
cool, but...

crypto is not a

but a method to manage the attack surface

ATTACK SURFACE

– all the possible places where sensitive data may be stolen by adversary



https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet

it’s easier to monitor
the suspicious behavior
in a small place

HANDLING SECRET DATA WITH CARE

avoid plain text as possible

manage keys properly

decrease time of plaintext secrets in memory

log, monitor and inspect

– HOW TO MANAGE THE ATTACK SURFACE OF MY DATA?

symmetric encryption
with poor key management

one container **one key**

one container one key

symmetric encryption
with poor key management

key leaked
→ data leaked

attack surface
is arbitrary

WHAT IS A CRYPTO-SYSTEM

Cryptosystem

From Wikipedia, the free encyclopedia

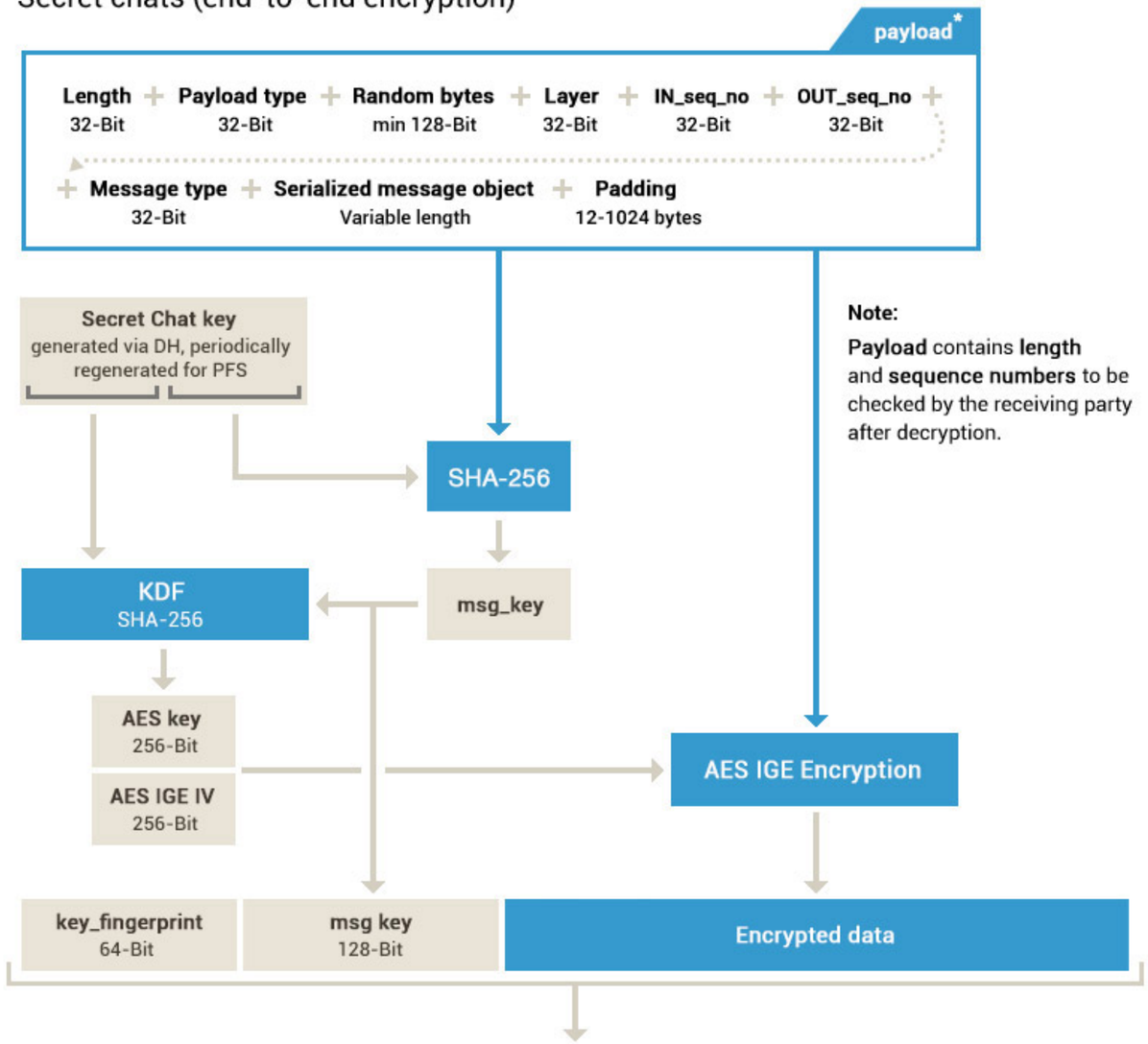
In cryptography, a **cryptosystem** is a suite of cryptographic algorithms needed to implement a particular security service, most commonly for achieving confidentiality (encryption).[1]

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

KEY AND TRUST MANAGEMENT SHOULD REFLECT YOUR SYSTEM

MTProto 2.0, part II

Secret chats (end-to-end encryption)



embedded into an outer layer of client-server (cloud) MTProto encryption, then into the transport protocol (TCP, HTTP, ..)

Important: After decryption, the receiver **must** check that msg_key = SHA-256(fragment of the secret chat key + decrypted data)

https://core.telegram.org/api/end-to-end

MESSAGING

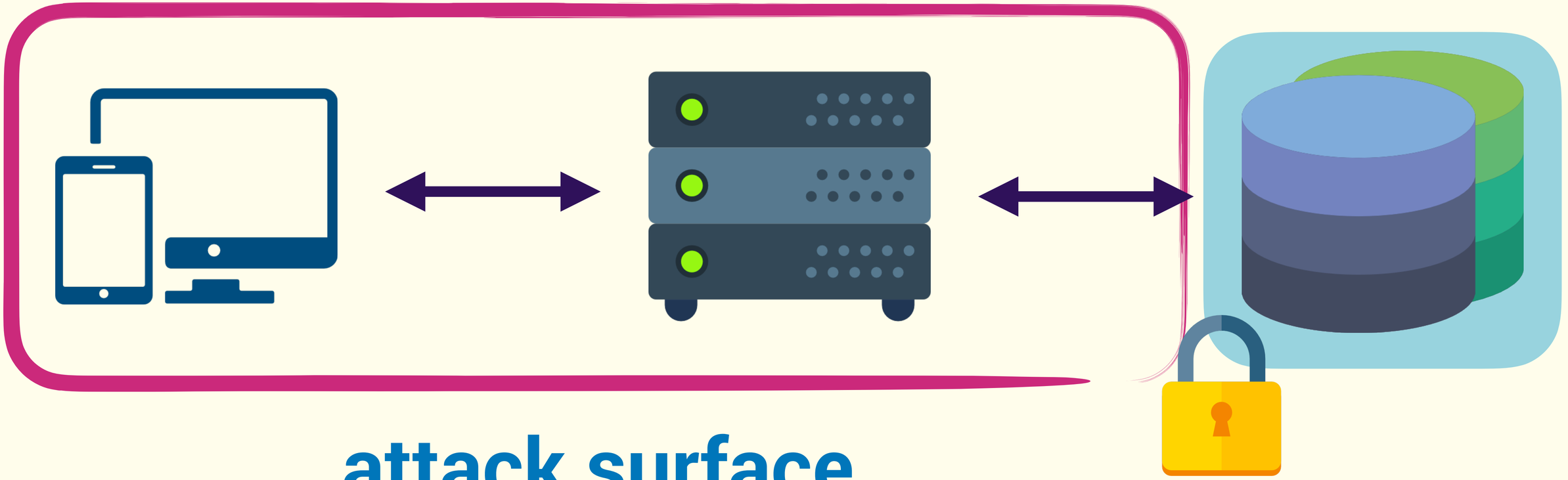
#qconlondon @vixentael

GOOD MESSAGING

IS E2EE

...but your infrastructures are not only for messaging

NAIVE DATABASE ENCRYPTION

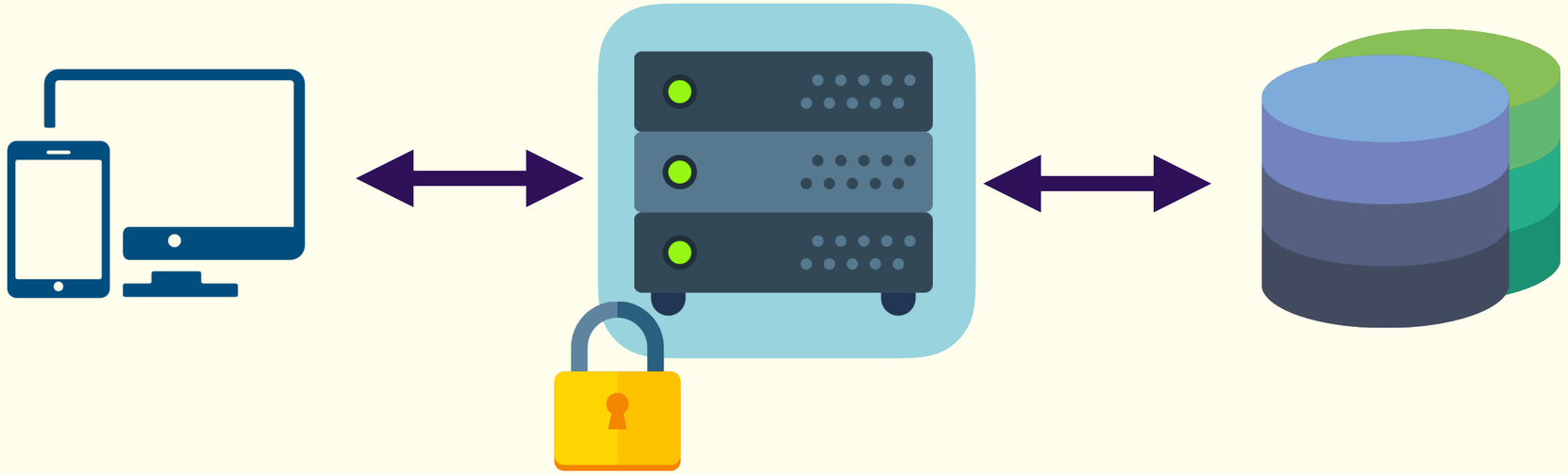


attack surface

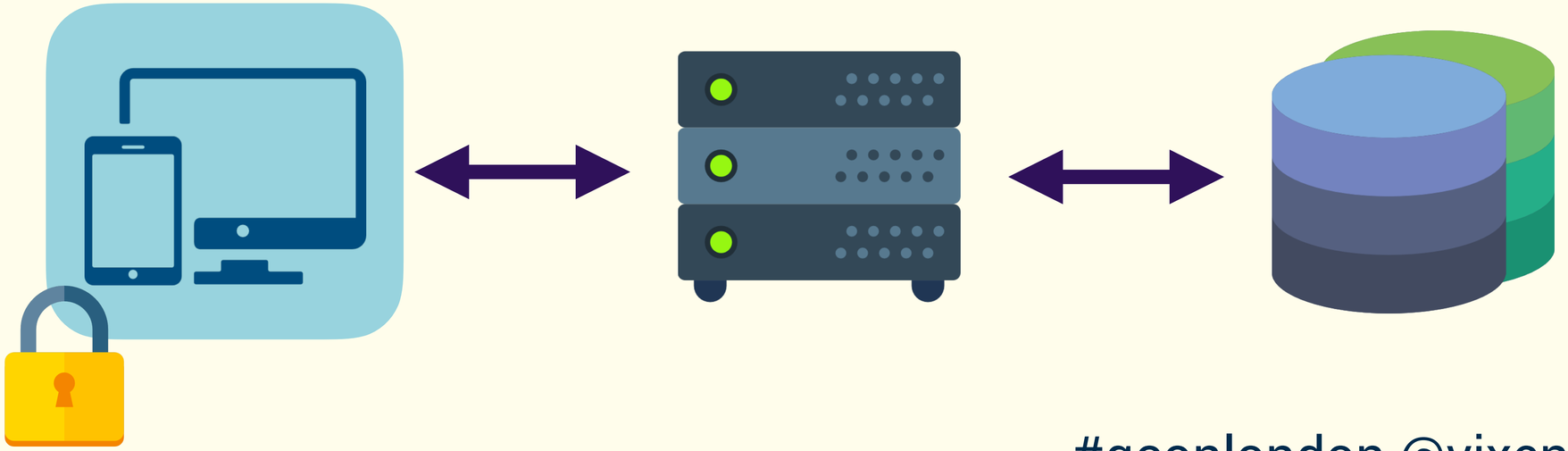
is almost everywhere

NARROWING ATTACK SURFACE

middleware-side encryption

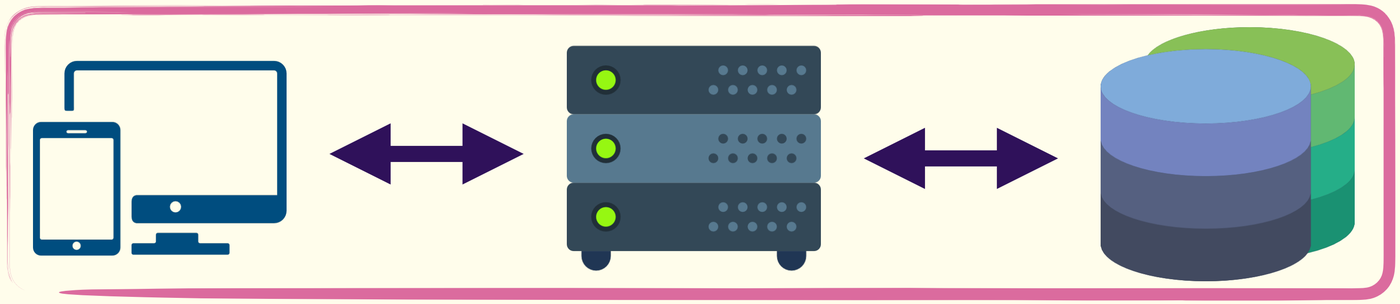


client-side encryption



MIDDLEWARE-SIDE ENCRYPTION

REAL-WORLD WEB SERVER



XSS

reflection attacks

MitM

SQL injections

code injections

crypto-miners everywhere

execution flow attacks

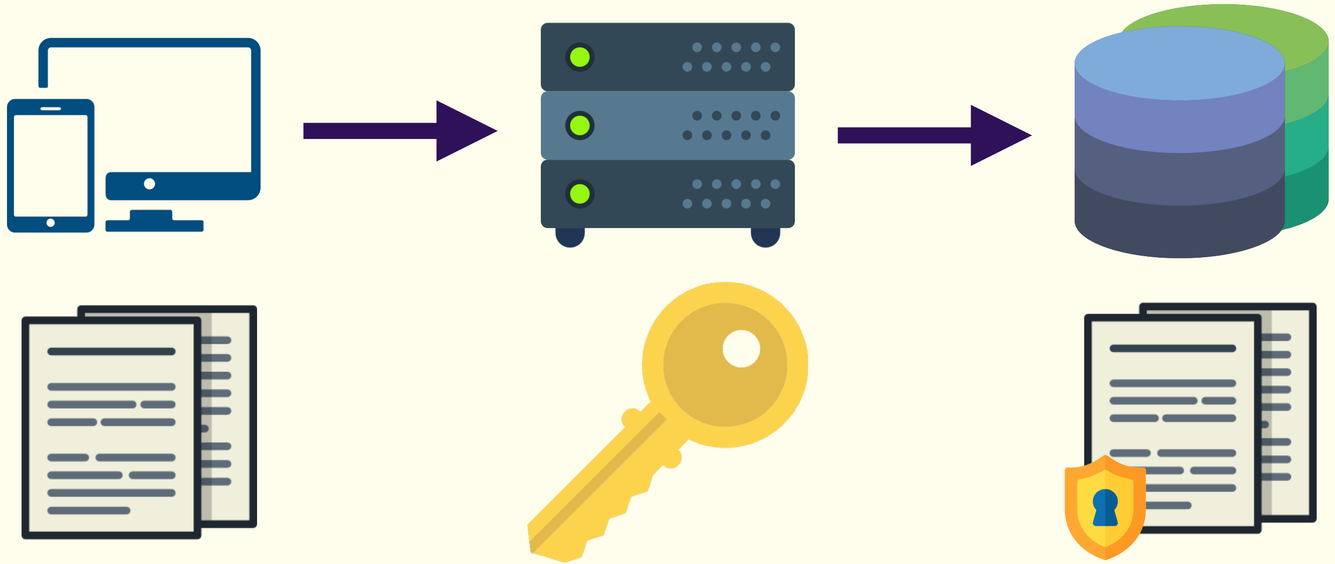
REAL-WORLD WEB SERVER

ATTACK SURFACE IS EVERYWHERE :(

monitor everything

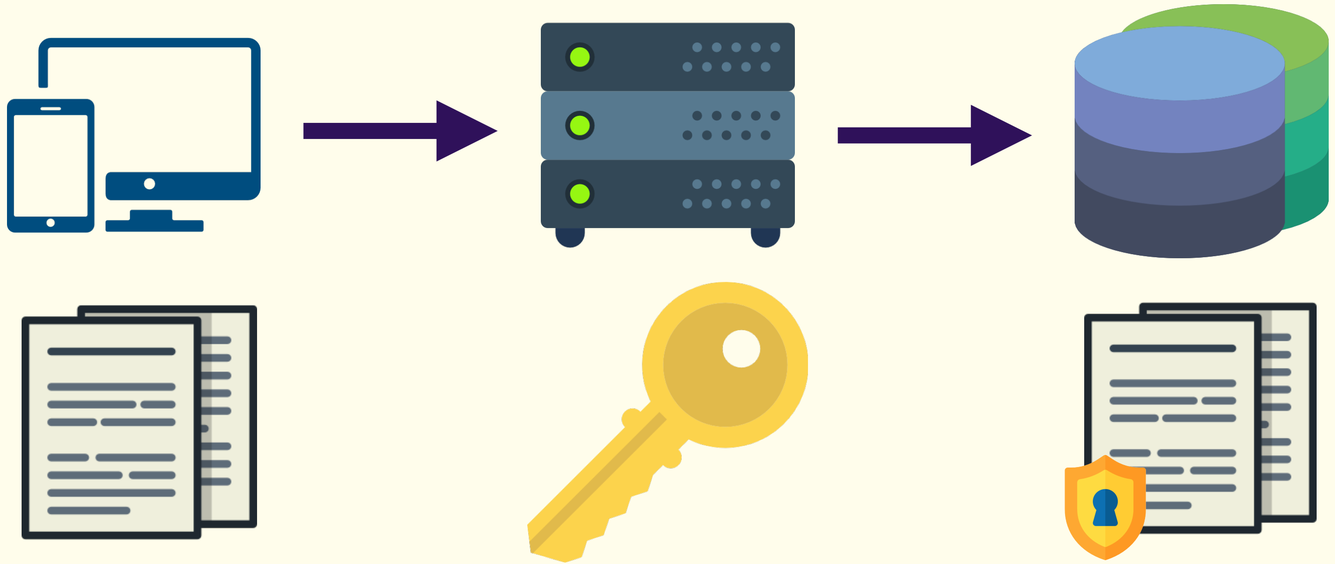
TRY SYMMETRIC ENCRYPTION?

encrypt/decrypt data
using symm key



TRY SYMMETRIC ENCRYPTION?

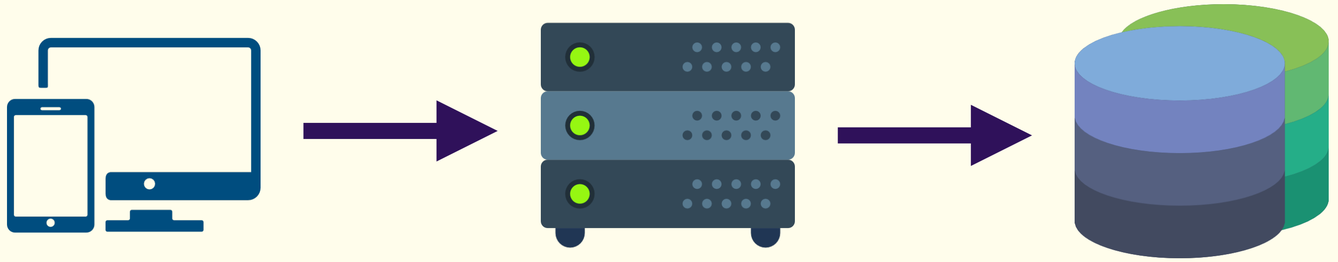
encrypt/decrypt data
using symm key



easy to steal a key

https://www.alibabacloud.com/help/faq-detail/37505.htm

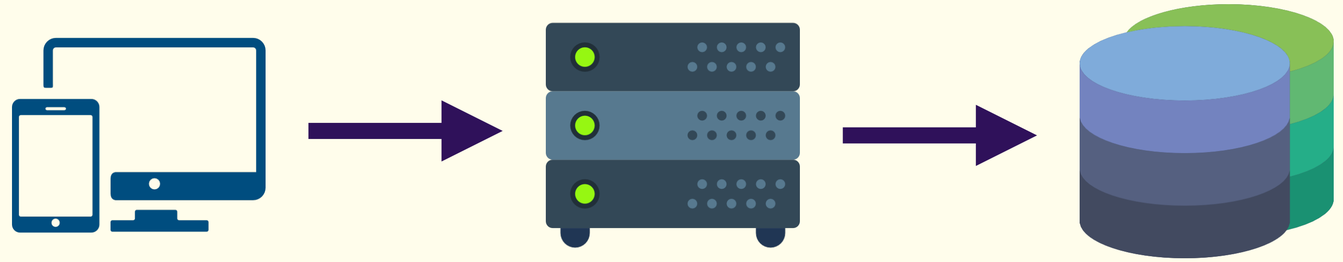
ASYMMETRIC ENCRYPTION



wrapped data = Enc(data, random symm key)

container = Enc(wrapped data, PK_{web}, $PubK_{tds}$)

ASYMMETRIC ENCRYPTION

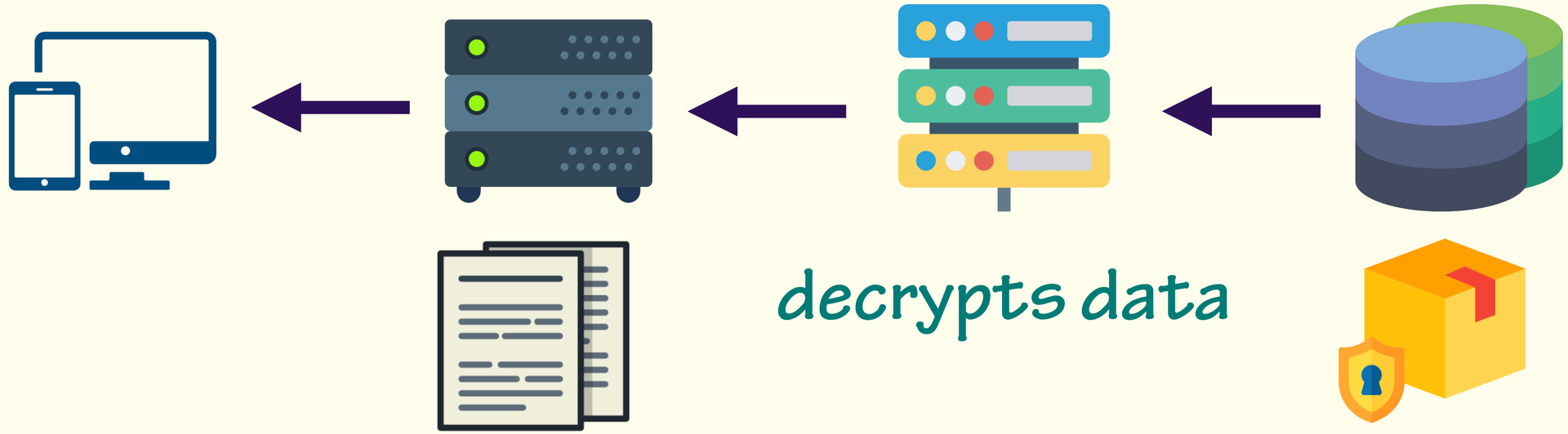


wrapped data = Enc(data, random symm key)

container = Enc(wrapped data, PK_{web}, $PubK_{tds}$)

PubKey of 'trusted decryption service'

TRUSTED DECRYPTION SERVICE

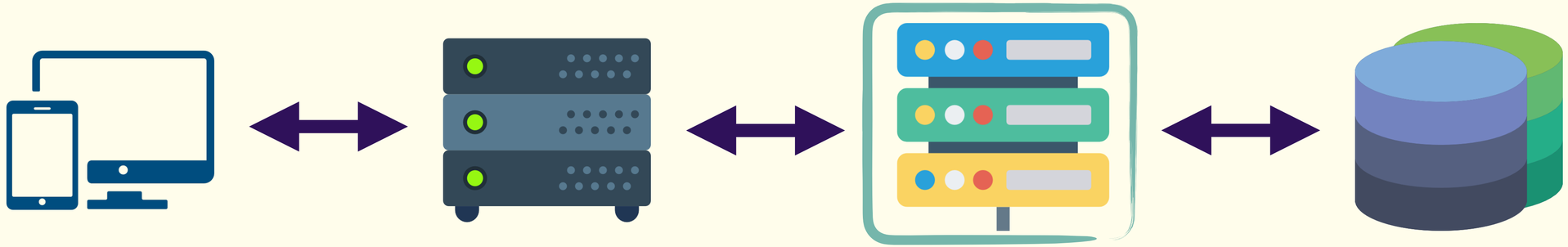


decrypts data

wrapped_data = Dec(container,
PK_{tds}, $PubK_{web}$)

data = Dec(wrapped_data,
random symm key)

SEPARATION OF DUTIES



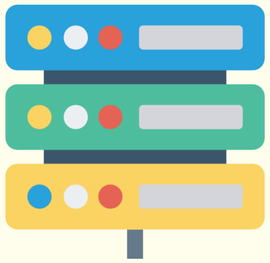
no decryption keys

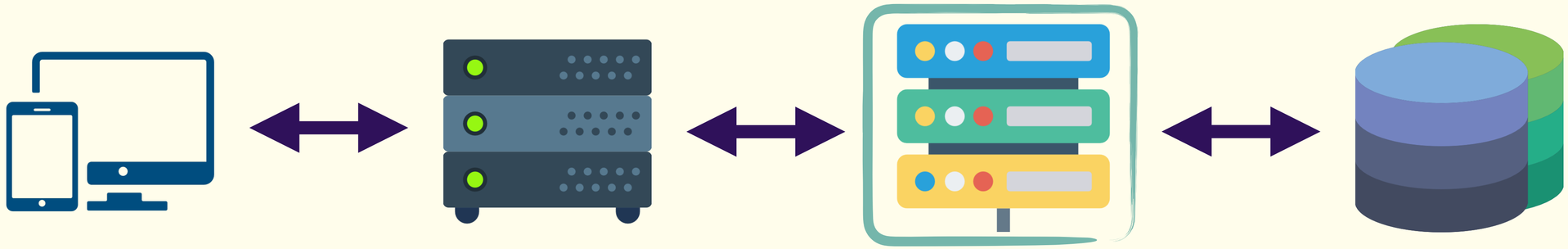
trusted element in infrastructure

monitor & log

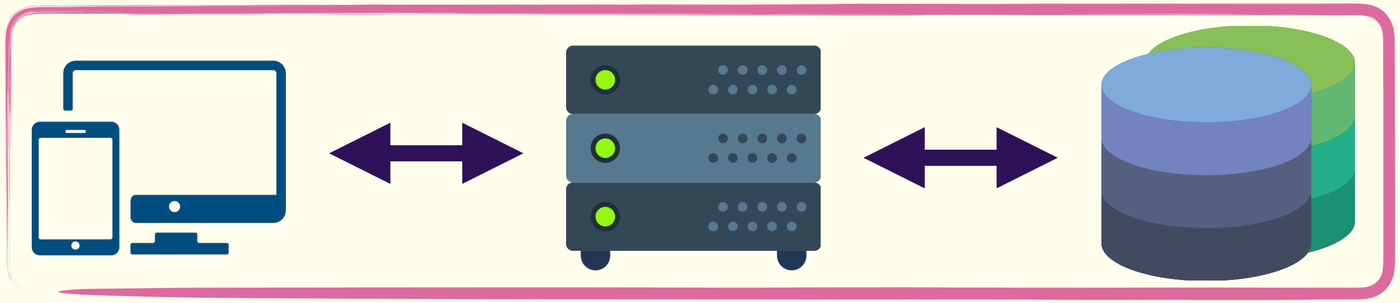
TRUSTED DECRYPTION SERVICE

NARROWED ATTACK SURFACE





monitor decryption proxy



monitor everything

WHERE TO USE THIS TECHNIQUE?

micro-services infrastructure

public-oriented interfaces

non trusted client side (browsers, IoT devices)

hard to store keys securely

HOW TO IMPLEMENT?

ACRA

https://github.com/cossacklabs/acra

HEXATIER

http://www.hexatier.com/

GREEN SQL

https://github.com/larskanis/greensql-fw

ORACLE DATABASE FIREWALL / TDE

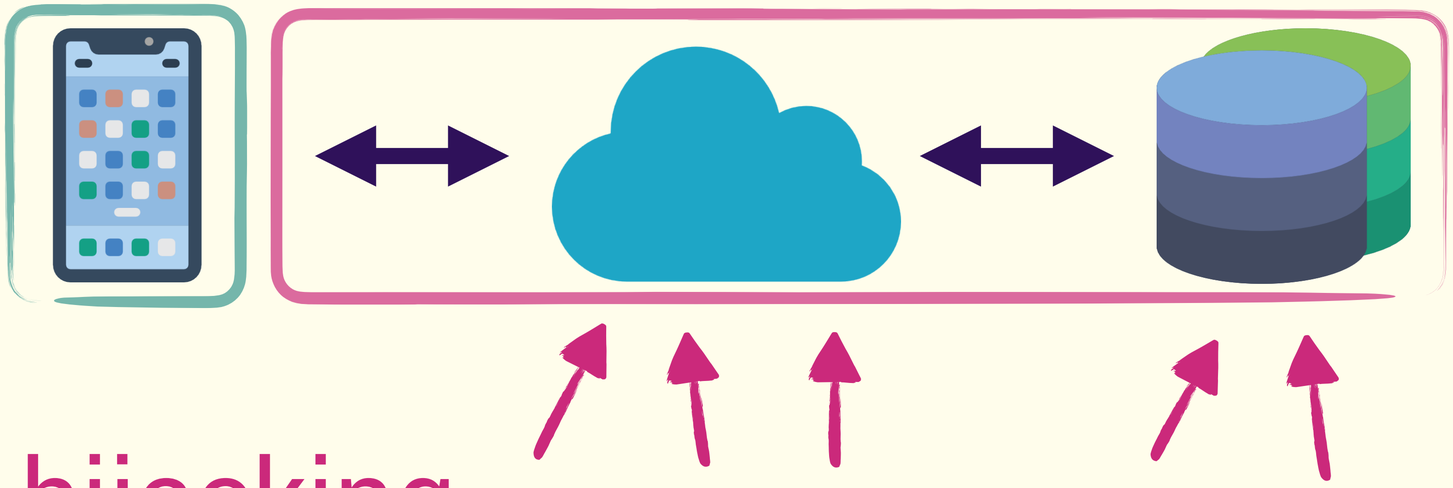
http://www.oracle.com/


4GIFs.com

CLIENT-SIDE ENCRYPTION

MOVE TRUST TO CLIENTS

trusted element in infrastructure



session hijacking

MitM

replay attacks

unattended backups

misconfigured ACL

P2P TRUST

user-generated keys

encrypted containers

system doesn't know anything about data

ZERO KNOWLEDGE ARCHITECTURES

#qconlondon @vixentael

ZKA is a design principle that enables software to provide services over protected client data without having an unencrypted access to it.

ZKA INCLUDES:

e2ee clients

ZKA INCLUDES:

e2ee clients

all operations are on encrypted data:

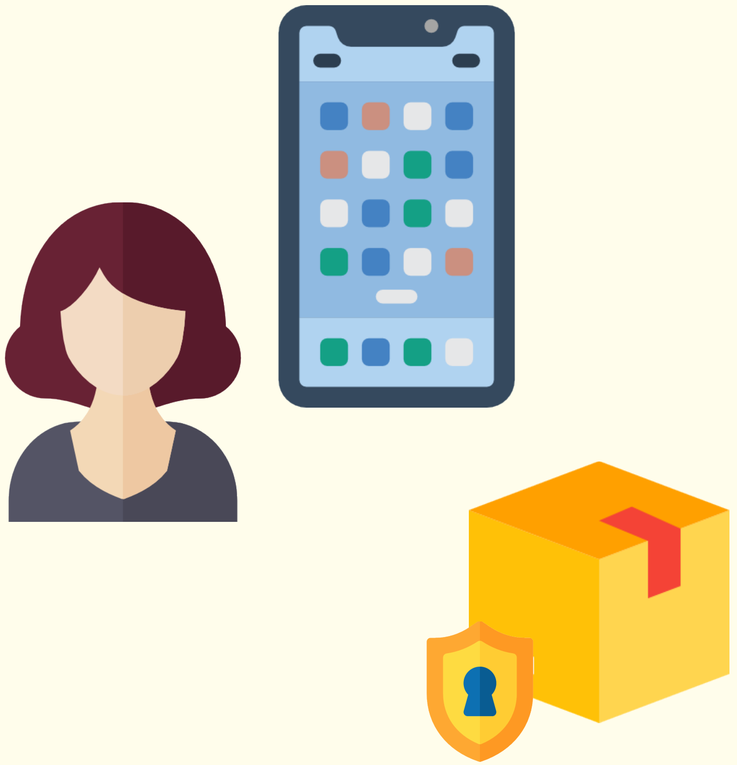
- CRUD
- control access to data from different users
- search (in encrypted data)

RISKS FOR ZKA:

weak key management

algorithm weakness

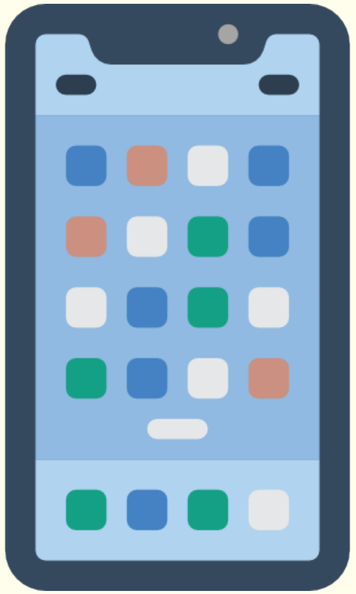
user pocket



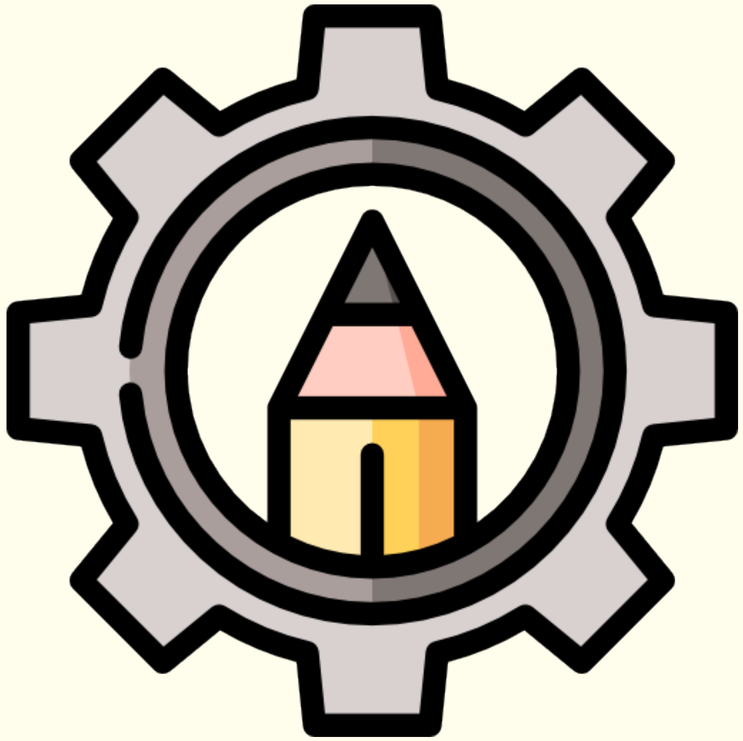
attack surface

WHEN TO USE ZKA?

trusted client side (mobile, HSM/TPM)



ZKA is already solved for specific use-cases or in a naive ways



MESSAGING

END-TO-END ENCRYPTION

AUTHENTICATION

ZERO KNOWLEDGE PROOF

https://www.cossacklabs.com/zero-knowledge-protocols-without-magic.html

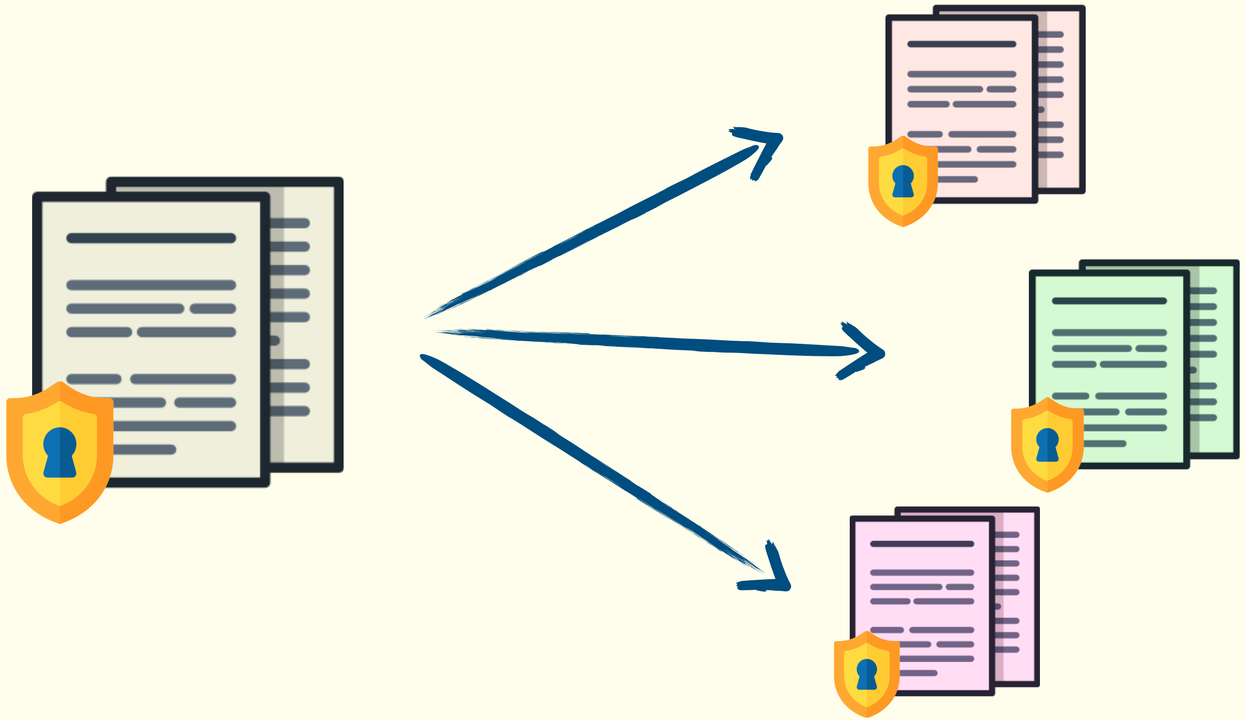
COLLABORATING ON DATA

???

– store encrypted
– share with others
– manage access to parties

SHARING ENCRYPTED DATA

naive approach



– duplications

– key management problems

OUR TAKE

give access to certain blocks of data to exact users

https://github.com/cossacklabs/hermes-core

HOW TO BUILD IT?

– Key wrapping

blocks

storage keys

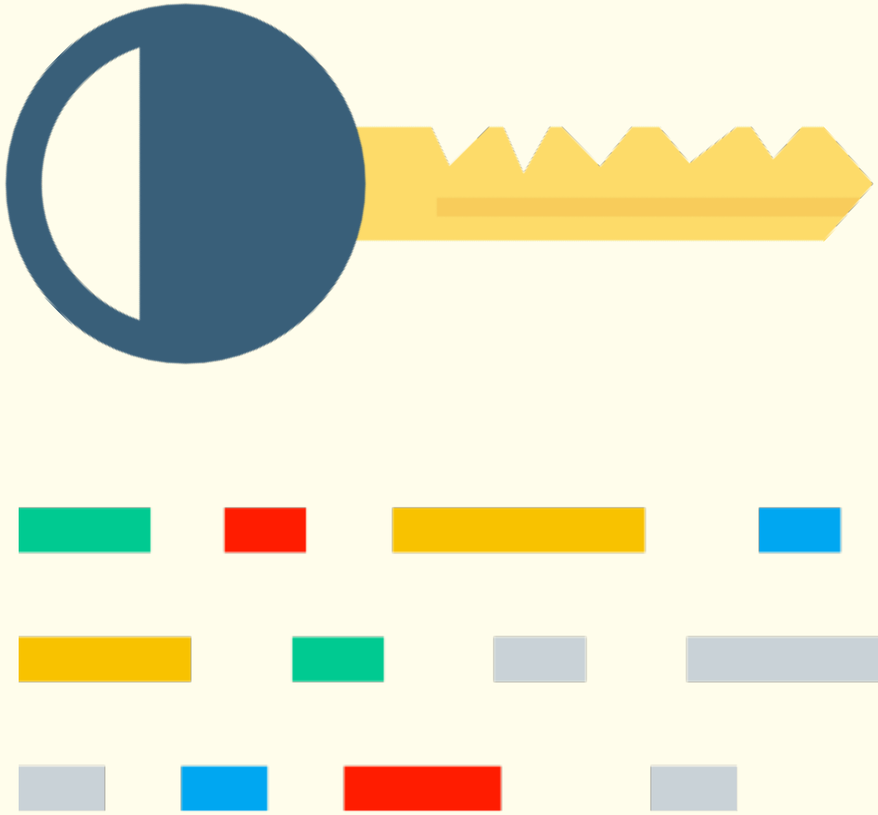
user keys

HOW TO BUILD IT?

- Key wrapping
- Manage privileges

HOW TO BUILD IT?

- Key wrapping
- Manage privileges
- Control requests



MORE POSSIBLE USE-CASES

shared
audit logs

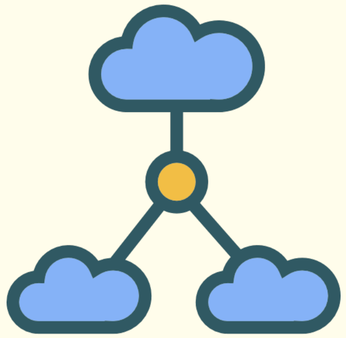
complex docs,
spreadsheets

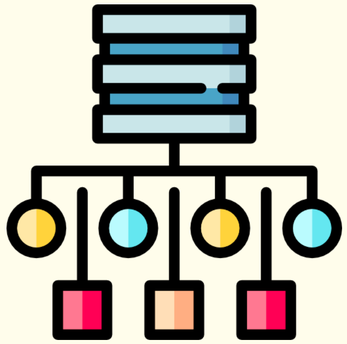
file system

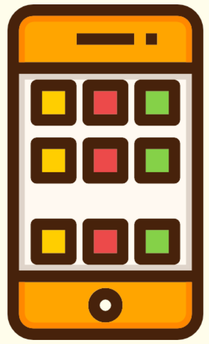
document store

config files

protection







OTHER IMPLEMENTATIONS

HERMES

https://github.com/cossacklabs/hermes-core

ZEROKIT

https://tresorit.com/zerokit

LAFS

https://tahoe-lafs.org/trac/tahoe-lafs

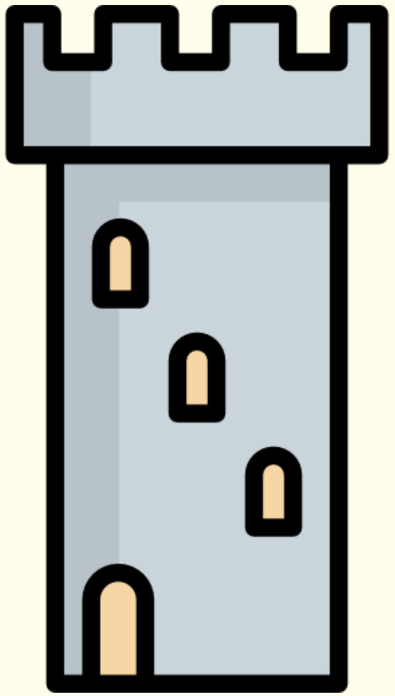
monitor client side

monitor everything

MORE GOODIES TO THINK ABOUT

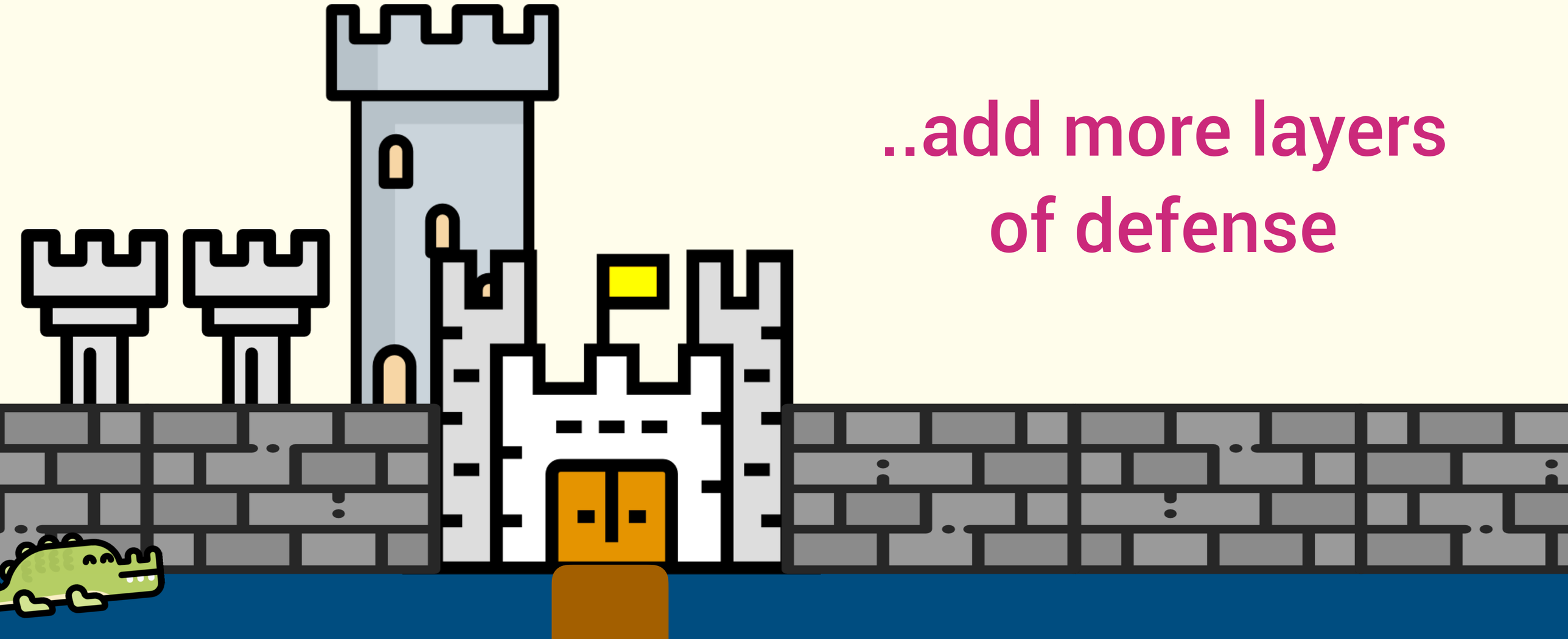
Cryptography is well implemented,
if it allows to narrow attack surface,
and increase control of data.

ECHELONIZATION



if the system has
one perimeter,
it will fail!

ECHELONIZATION



..add more layers of defense

EXCEPT CRYPTO, YOU ALSO NEED

log and monitor events

intrusion pattern detection

access control

firewall

...

Why does cryptographic software fail?
A case study and open problems

David Lazar, Haogang Chen, Xi Wang, and Nickolai Zeldovich
MIT CSAIL

269 CVEs
from 2011-2014

17% **bugs inside crypto libs**

83% **misuses of crypto libs**
by individual apps

https://pdos.csail.mit.edu/papers/cryptobugs:apsys14.pdf

RECAP

2

THINGS TO REMEMBER

1. cryptography aims to narrow the attack surface
2. choose relevant encryption scheme
3. combine crypto and classic techniques
4. there is a lib for that


DON'T ROLL
YOUR OWN
CRYPTO

LINKS

12 and 1 ideas how to enhance backend data security
https://medium.com/@cossacklabs/12-and-1-ideas-how-to-enhance-backend-data-security-4b8ceb5ccb88

Explain Like I'm 5: Zero Knowledge Proof
https://hackernoon.com/eli5-zero-knowledge-proof-78a276db9eff

DevOps and security: from trenches to command centers
https://medium.com/@9gunpi/devops-and-security-from-trenches-to-command-centers-466dfb58fe5b

How GDPR Will Change The Way You Develop
https://www.smashingmagazine.com/2018/02/gdpr-for-web-developers/

MY OTHER SECURITY SLIDES

DON'T WASTE TIME ON LEARNING CRYPTOGRAPHY: BETTER USE IT PROPERLY

KEYS FROM THE CASTLE

ANCIENT ART OF MANAGING KEYS AND TRUST

ZERO KNOWLEDGE ARCHITECTURES

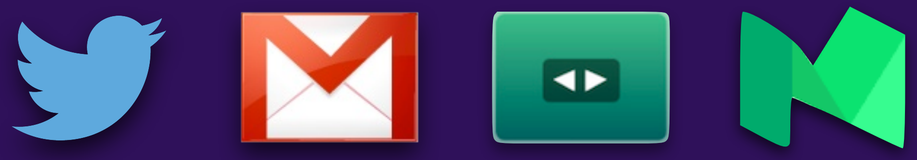
for mobile applications

Building user-centric security model in iOS apps

https://github.com/vixentael/my-talks

...and more

@vixentael



Feel free to reach me with security questions.

I do check my inbox :)

Product Engineer

COSSACK LABS

IMAGE CREDITS

www.flaticon.com

Authors:

freepik, linector, switficons, pixelperfect, smashicons, icon pond, dinosoftlabs