intel

Secure Banking Transaction System

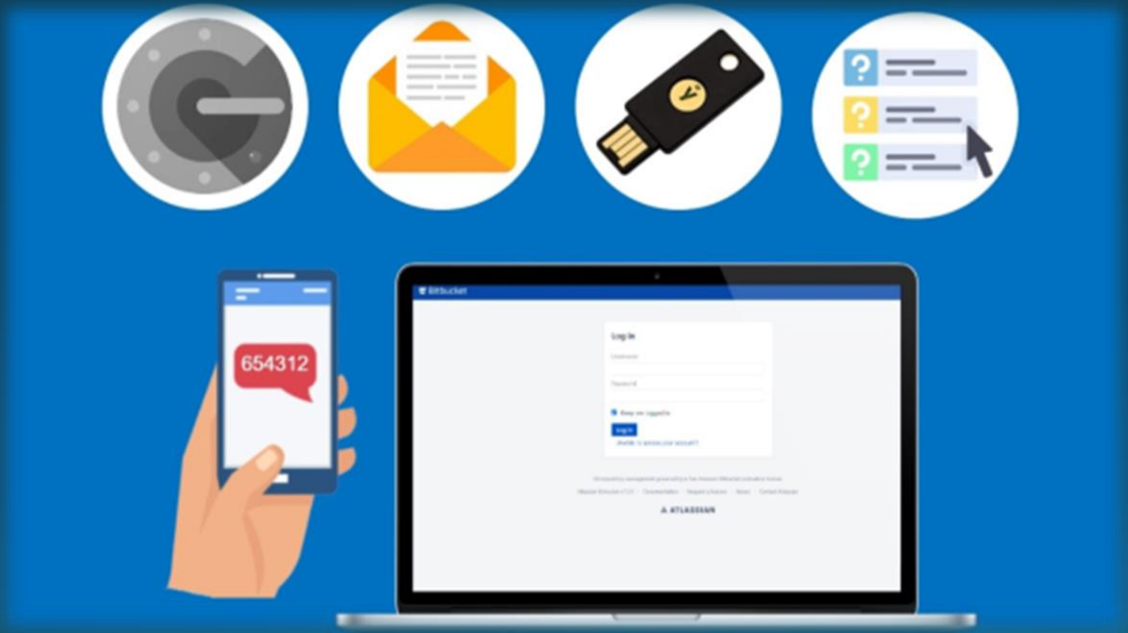
By: Shachar Markovich
Naor Maman
Tzuf Newfeld

Today:

- About the Project:
  - The Problem
  - Our Solution
- Technical details
- Future Work
- Demonstration & Code

The Problem

SMS, Authenticator App, Security Key
are NOT secure enough



Our Solution

Motivations:

- Securing the communication with servers
- 2FA not secure enough
- Ensure that only real person can enter the OTP

The Solution:

- Using **Intel DAL** for MFA

Technical details

How it works:

- Keys exchange on Init
- OTP secret creation on registration
- Essential data send in logging
- 2FA before critical operations

Messages format:

OpCode|Data as json

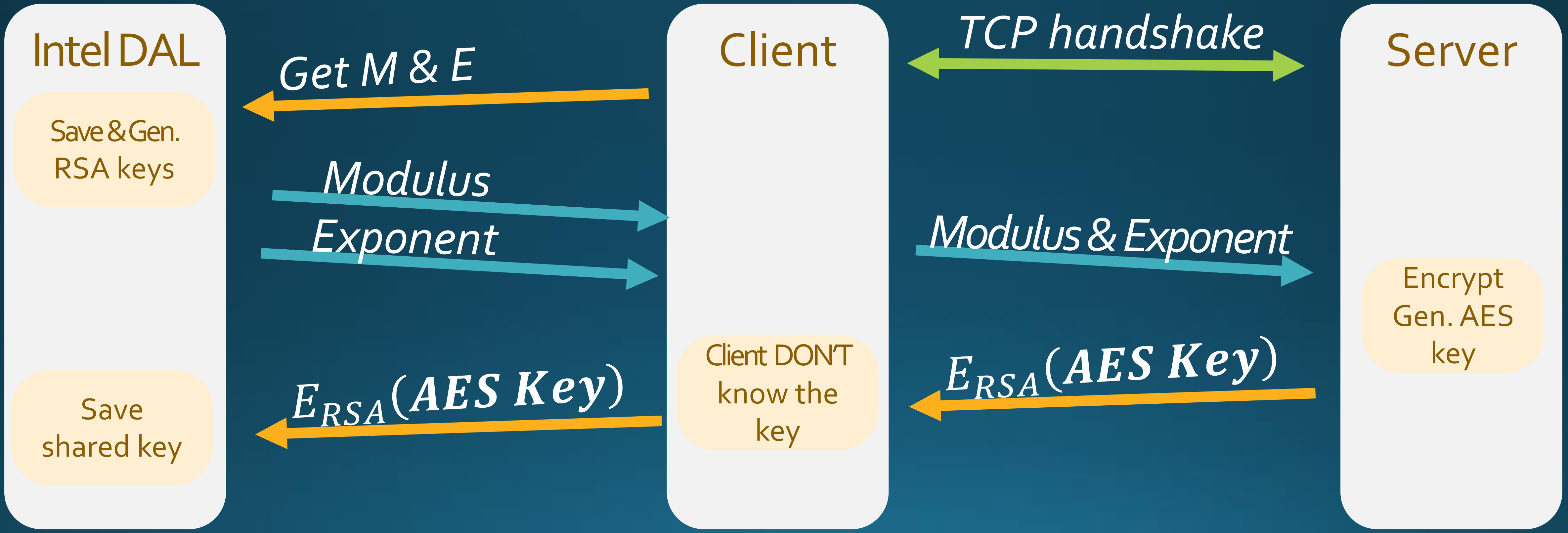
Technical details

Encryption:

- Long & Short-Term keys
- Intel Dal perform Encryptions
- Intel Dal save sensitive data

Technical details

Key Exchange Flow



Intel DAL

Client

Server

TCP handshake

Get M & E

Save & Gen. RSA keys

Modulus

Exponent

Modulus & Exponent

Encrypt Gen. AES key

$E_{RSA}(\boldsymbol{AES\ Key})$

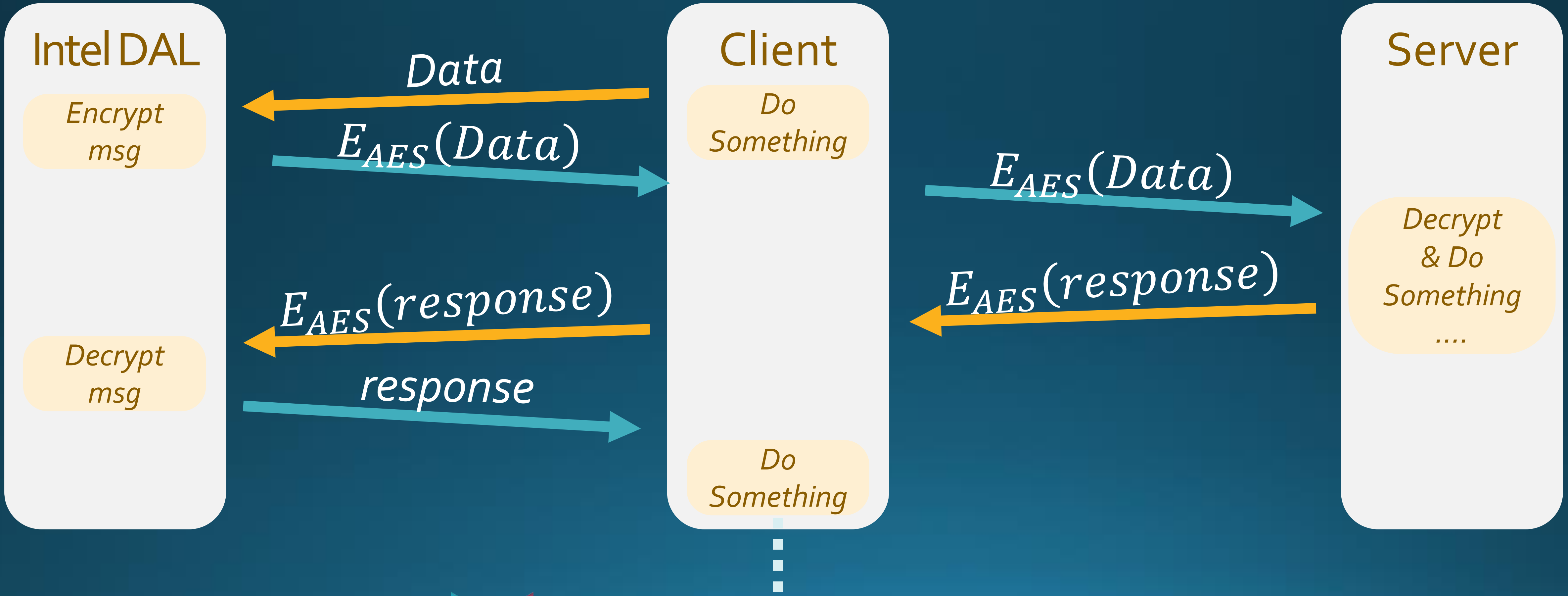
Client DON'T know the key

$E_{RSA}(\boldsymbol{AES\ Key})$

Save shared key

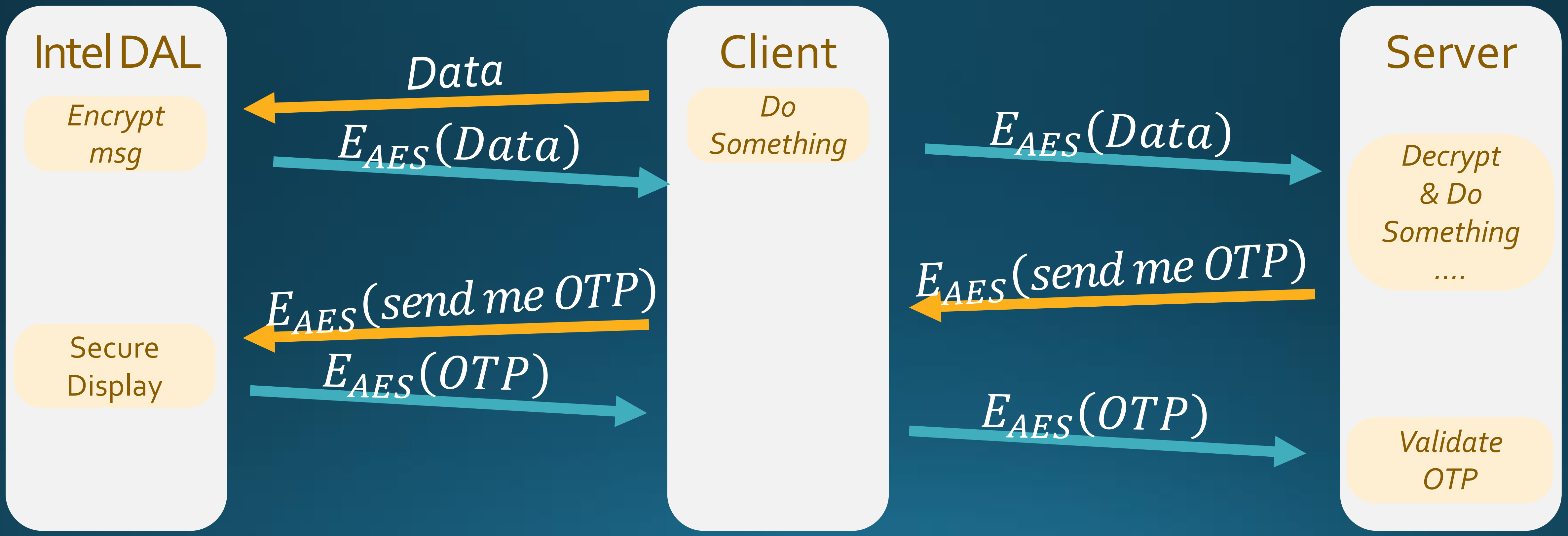
Technical details

Communication Flow



Technical details

2FA Flow



Intel DAL

Encrypt msg

Secure Display

Data

$E_{AES}(Data)$

$E_{AES}(send\ me\ OTP)$

$E_{AES}(OTP)$

Client

Do Something

$E_{AES}(Data)$

$E_{AES}(send\ me\ OTP)$

$E_{AES}(OTP)$

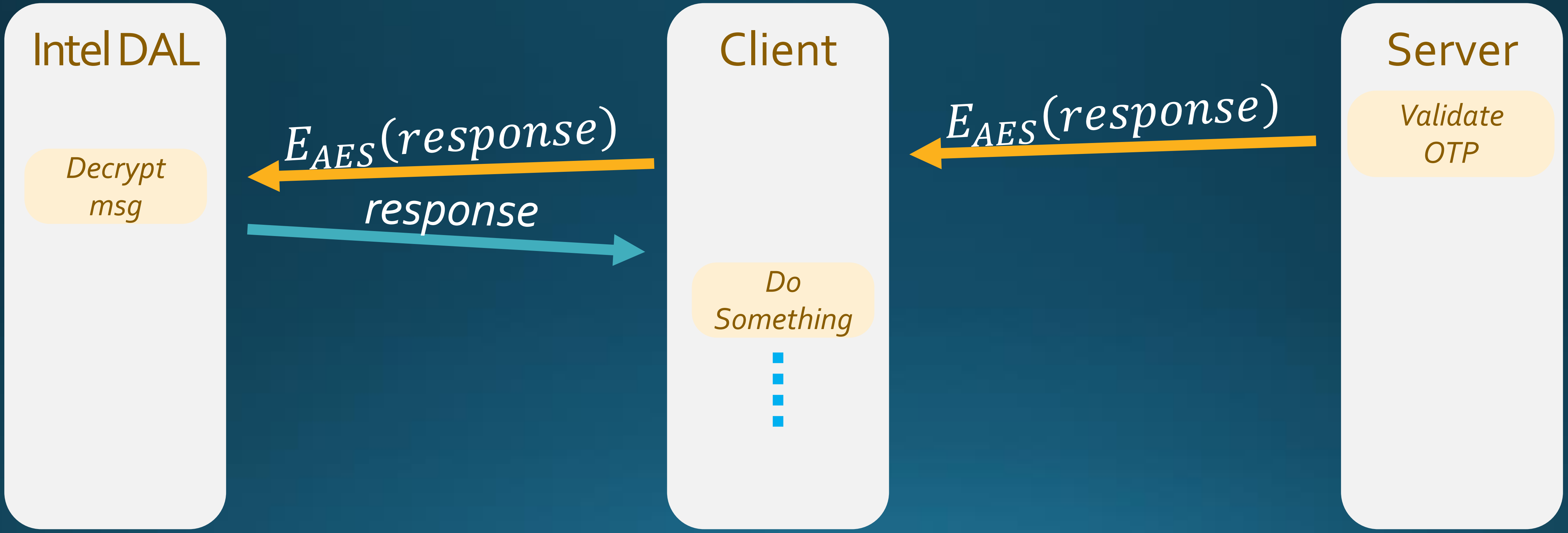
Server

Decrypt & Do Something ….

Validate OTP

Technical details

2FA Flow



Intel DAL

Decrypt msg

$E_{AES}(response)$

response

Client

Do Something

⋮

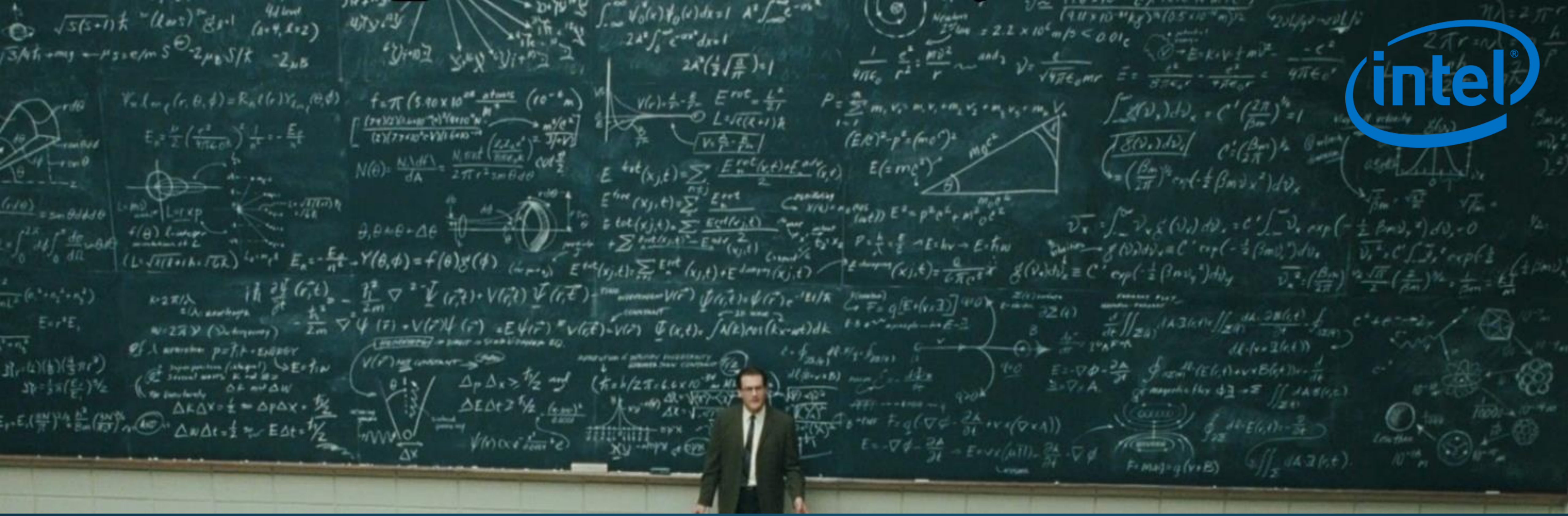
$E_{AES}(response)$

Server

Validate OTP

Future Work

- Immobility
- Combine with existing 2FA
- Parallel Server
- More Validations in server side
- D.H. Key Exchange



Questions?

Shachar Markovich & Naor Maman & Tzuf Newfeld

Thank you!