[redacted]

TRAIL OF BITS

Fuck RSA

Summercon 2019
Ben Perez

RSA

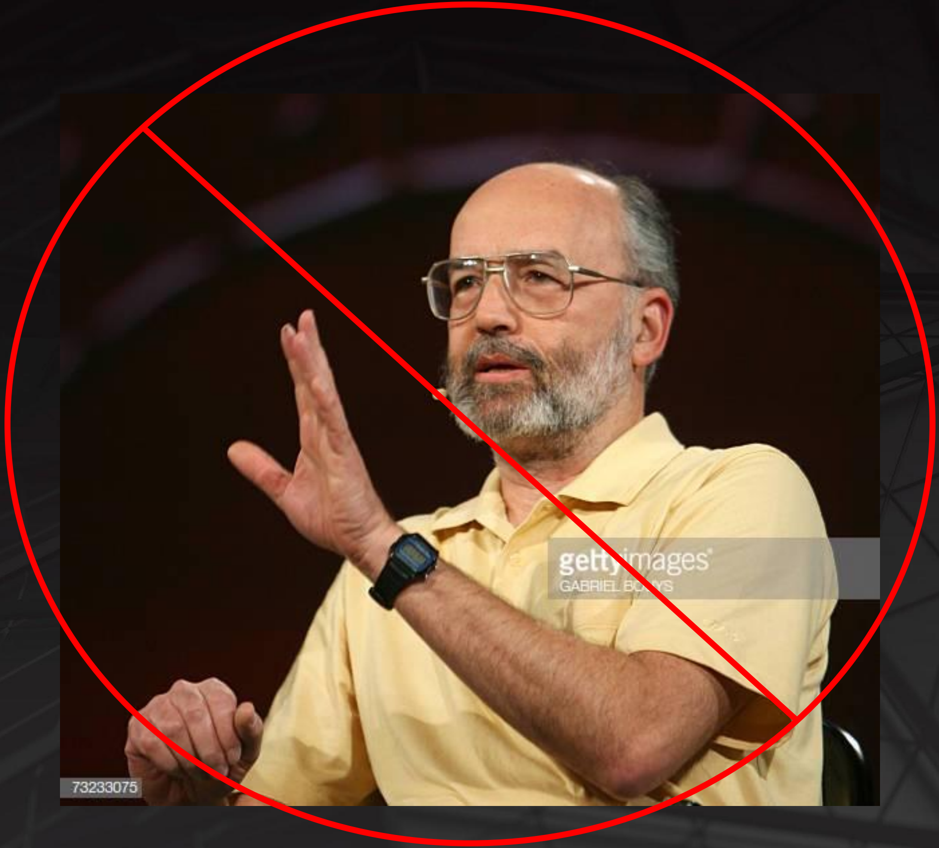
🖕🖕🖕

TRAIL OF BITS

Fuck RSA

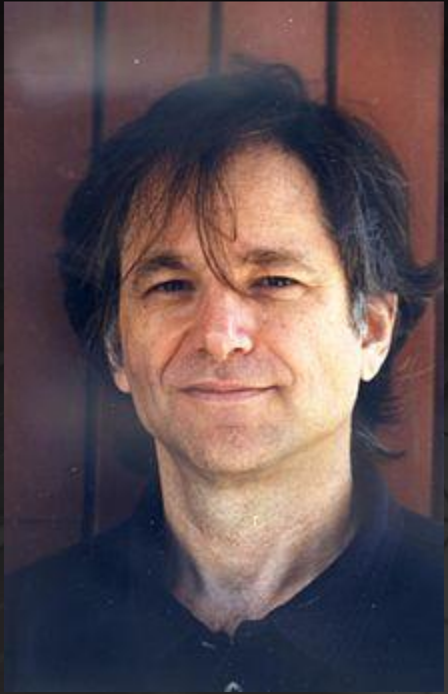
RSA
SecurID®
461520
Secured by RSA

Lizard Person



Cancelled

Biologist



RSA INVENTED
1977

Rivest Shamir Adleman Elvis

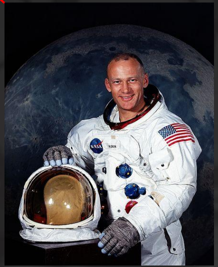
21 16 23 18

78 + 4096 - 2920 = 1254

RSA Key Size 2019

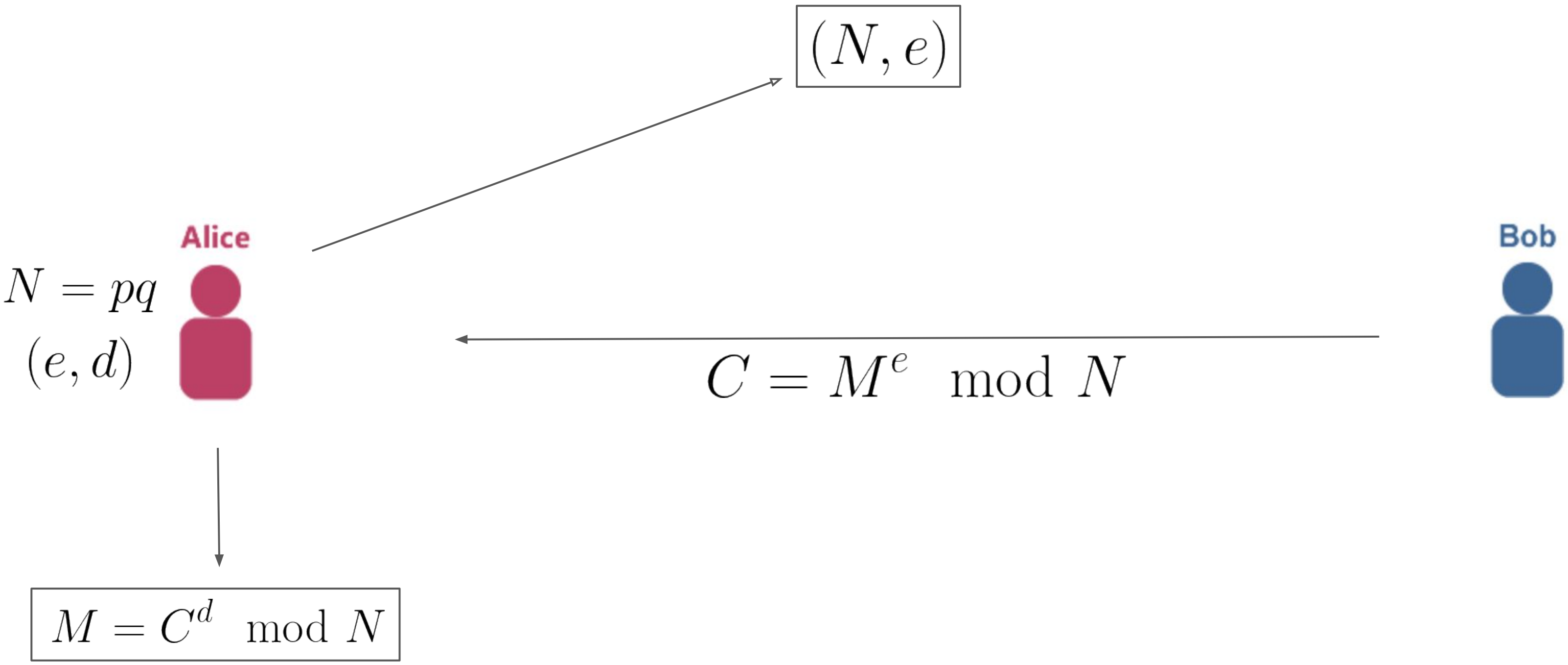
1254 / 2 + 39 =

6 6 6

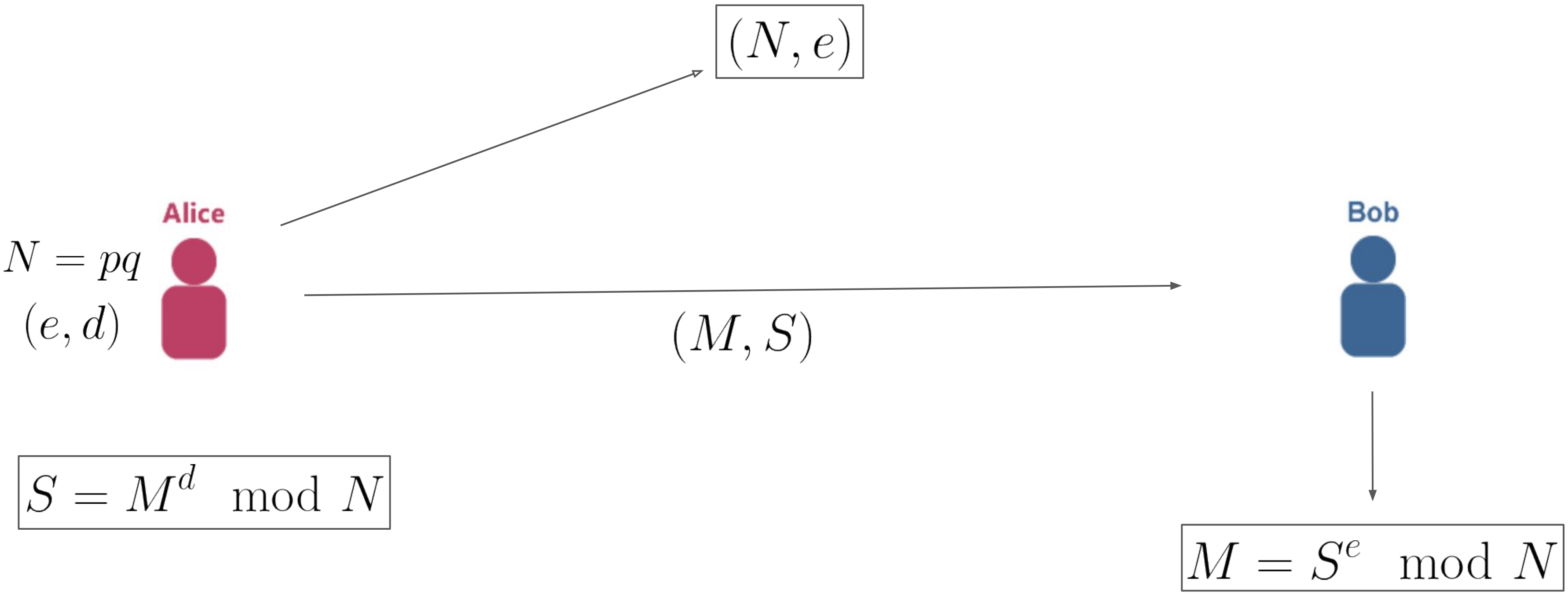


RSA Primer

What is RSA



What is RSA



$$\boxed{(N, e)}$$

Alice

$N = pq$

(e, d)

(M, S)

Bob

$$\boxed{S = M^d \mod N}$$

$$\boxed{M = S^e \mod N}$$

Parameter Selection

Parameter Selection

$$M = (M^e)^d \mod pq$$

Primes

$$M = (M^e)^d \mod \underline{pq}$$

If Alice reuses p for another RSA modulus pq’, attacker can factor using GCD

Primes

$$M = (M^e)^d \mod \underline{pq}$$

If Alice reuses p for another RSA modulus pq’, attacker can factor using GCD

If p and q share approximately half of their upper bits, then pq can be factored using Fermat’s method

Primes

$$M = (M^e)^d \mod \underline{pq}$$

If Alice reuses p for another RSA modulus pq’, attacker can factor using GCD

If either p or q contains too many contiguous zero bits, then pq can be factored using Coppersmith’s method

If p and q share approximately half of their upper bits, then pq can be factored using Fermat’s method

Primes

$$M = (M^e)^d \mod \underline{pq}$$

If Alice reuses p for another RSA modulus pq’, attacker can factor using GCD

If either p or q contains too many contiguous zero bits, then pq can be factored using Coppersmith’s method

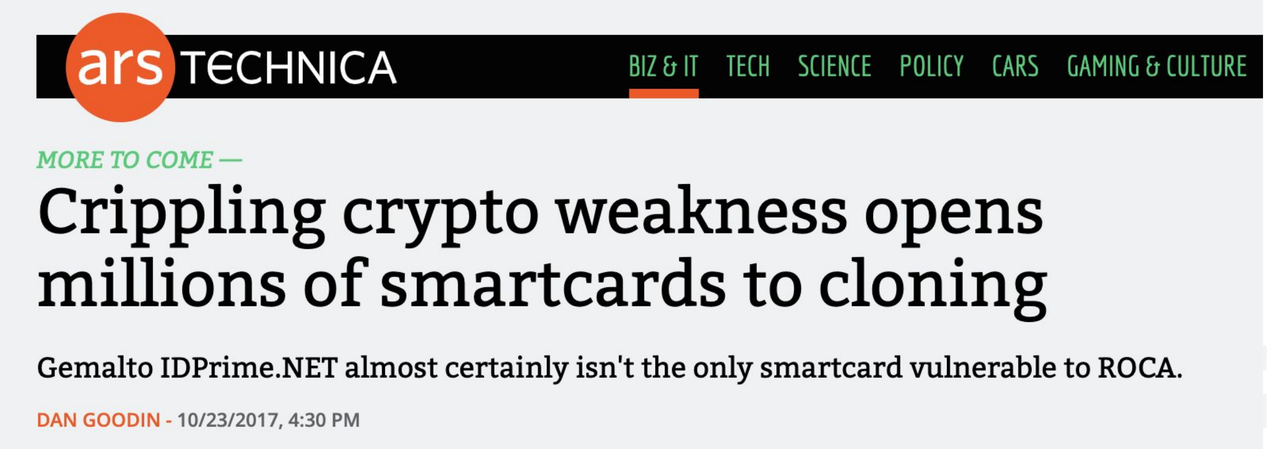
If p and q share approximately half of their upper bits, then pq can be factored using Fermat’s method

If p-1 or q-1 has small prime factors, then can use Pollard p-1 to factor pq

Primes

$$M = (M^e)^d \mod \underline{pq}$$

If Alice reus
RSA modul
can factor u



MORE TO COME —

Crippling crypto weakness opens millions of smartcards to cloning

Gemalto IDPrime.NET almost certainly isn't the only smartcard vulnerable to ROCA.

DAN GOODIN - 10/23/2017, 4:30 PM

If p ar … rime factors,
their u … to factor pq
factored using Fermat's method

Private Exponent

$$M = (M^e)^{\underline{d}} \mod pq$$

- Small private exponent speeds up decryption

Private Exponent

$$M = (M^e)^{\underline{d}} \mod pq$$

- Small private exponent speeds up decryption
- If d < ∜pq, then Eve can recover private key using continued fractions

Private Exponent

$$M = (M^e)^{\underline{d}} \mod pq$$

- Small private exponent speeds up decryption
- If d < ∜pq, then Eve can recover private key using continued fractions
- Can use Chinese remainder theorem to speed up decryption instead of picking small d - vulnerable to fault attacks.

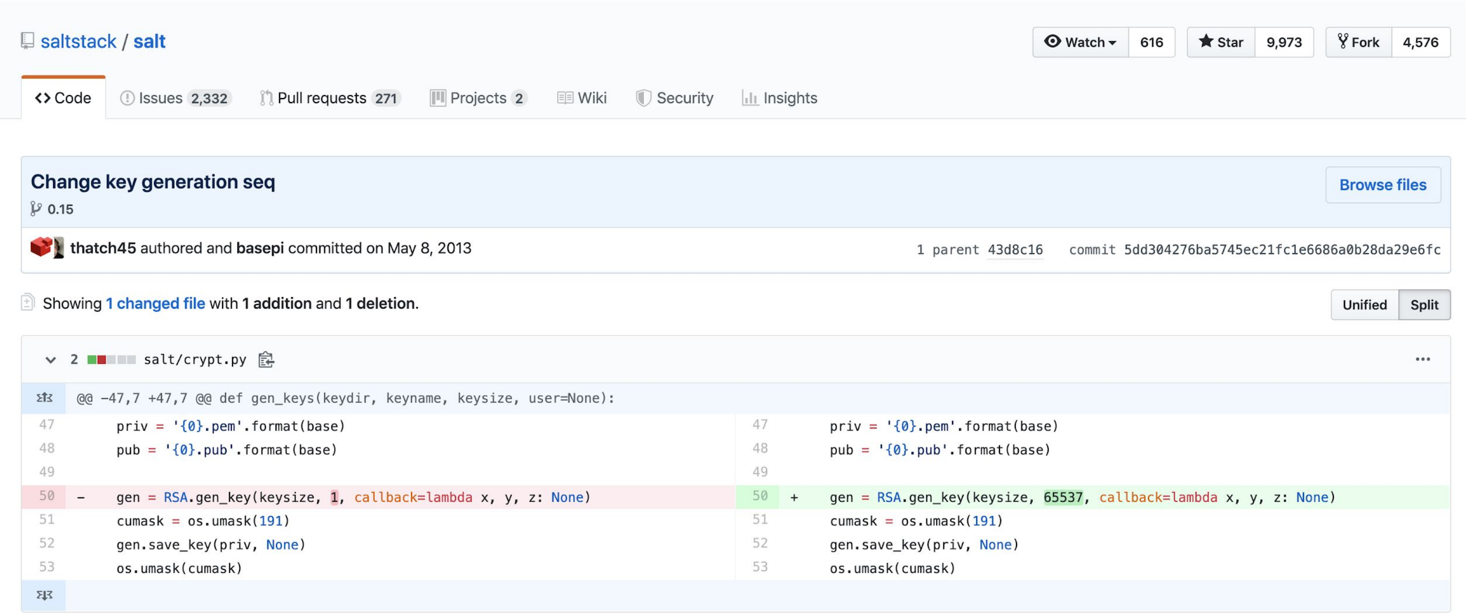
Public Exponent

$$M = (M^{\underline{e}})^d \mod pq$$

- Common to use e = 3, 17, 65537
- e = 3 is very bad
- Related messages can be decrypted
- Partial key exposure attack
- Signature forgery

How Bad is This IRL?

How Bad is This IRL?



Change key generation seq

0.15

thatch45 authored and **basepi** committed on May 8, 2013

`1 parent 43d8c16` `commit 5dd304276ba5745ec21fc1e6686a0b28da29e6fc`

Showing 1 changed file with **1 addition** and **1 deletion**.

2 `salt/crypt.py`

```
@@ -47,7 +47,7 @@ def gen_keys(keydir, keyname, keysize, user=None):
     priv = '{0}.pem'.format(base)
     pub = '{0}.pub'.format(base)

-    gen = RSA.gen_key(keysize, 1, callback=lambda x, y, z: None)
+    gen = RSA.gen_key(keysize, 65537, callback=lambda x, y, z: None)
     cumask = os.umask(191)
     gen.save_key(priv, None)
     os.umask(cumask)
```

How Bad is This IRL?

Developers should not need to understand algebraic number theory to build secure software

Padding Attacks

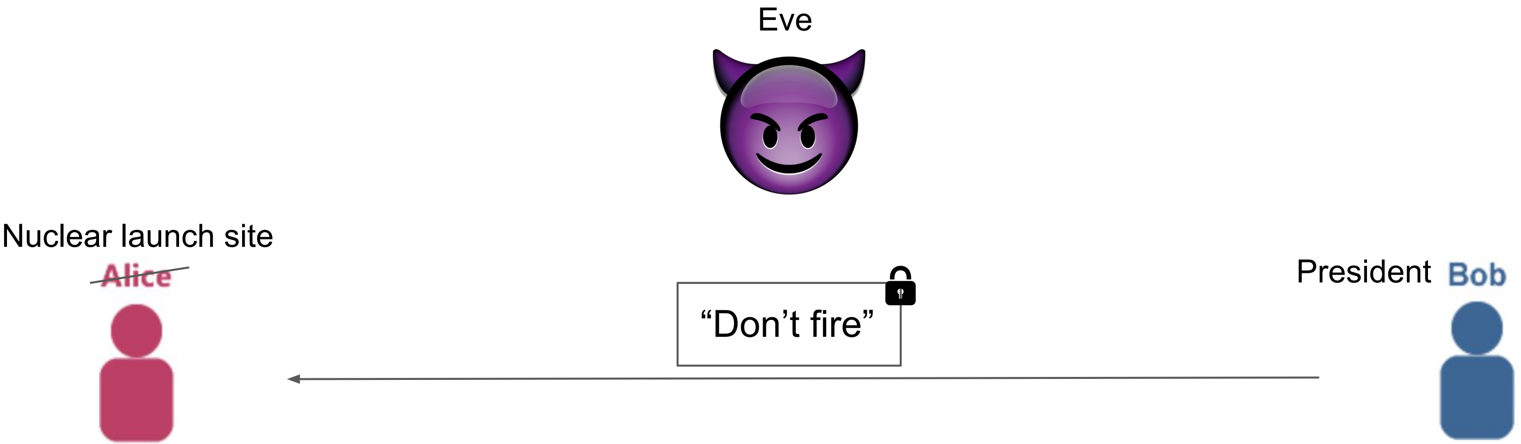
RSA Requires Padding

Nuclear launch site

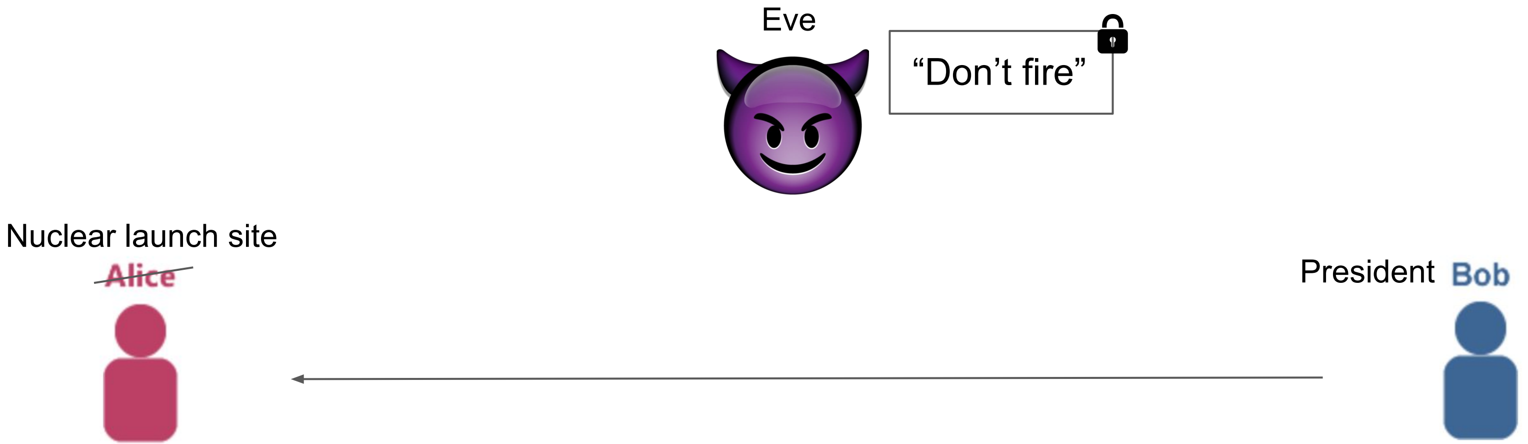
~~Alice~~

President Bob

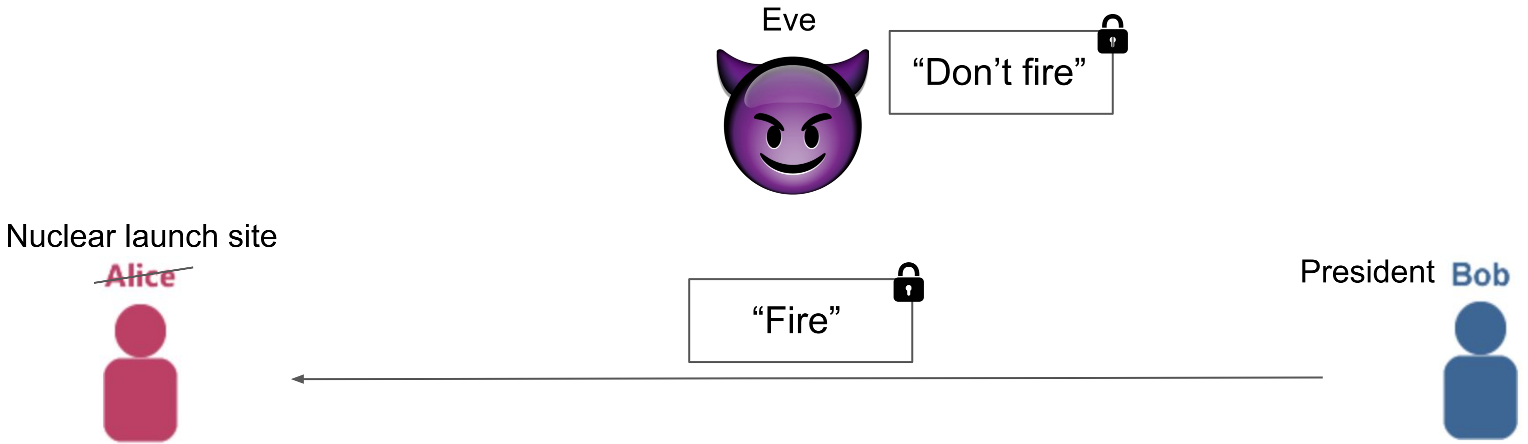
RSA Requires Padding



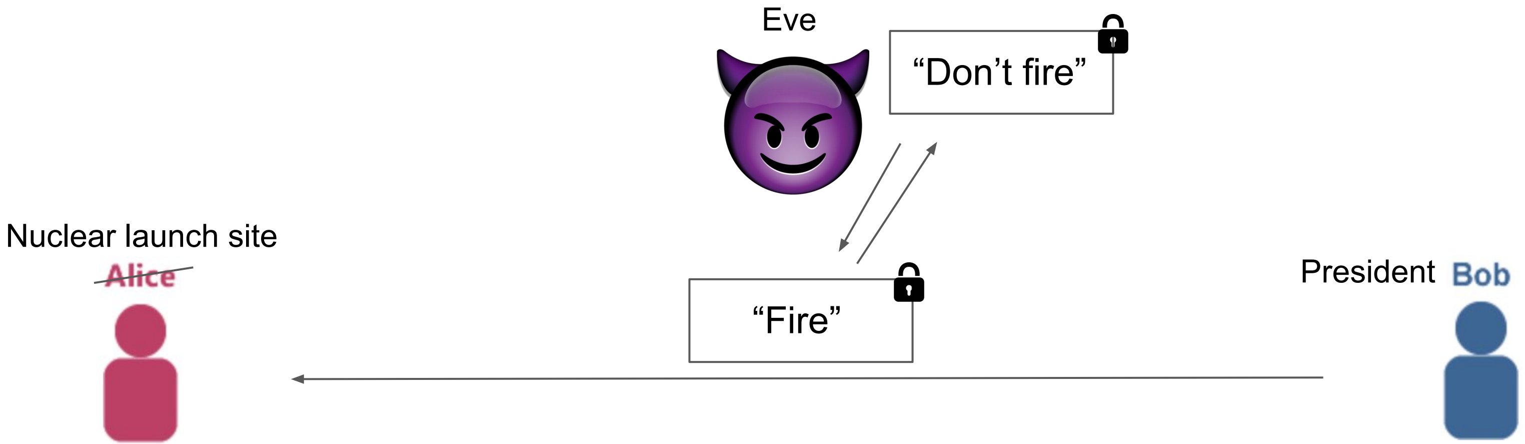
RSA Requires Padding



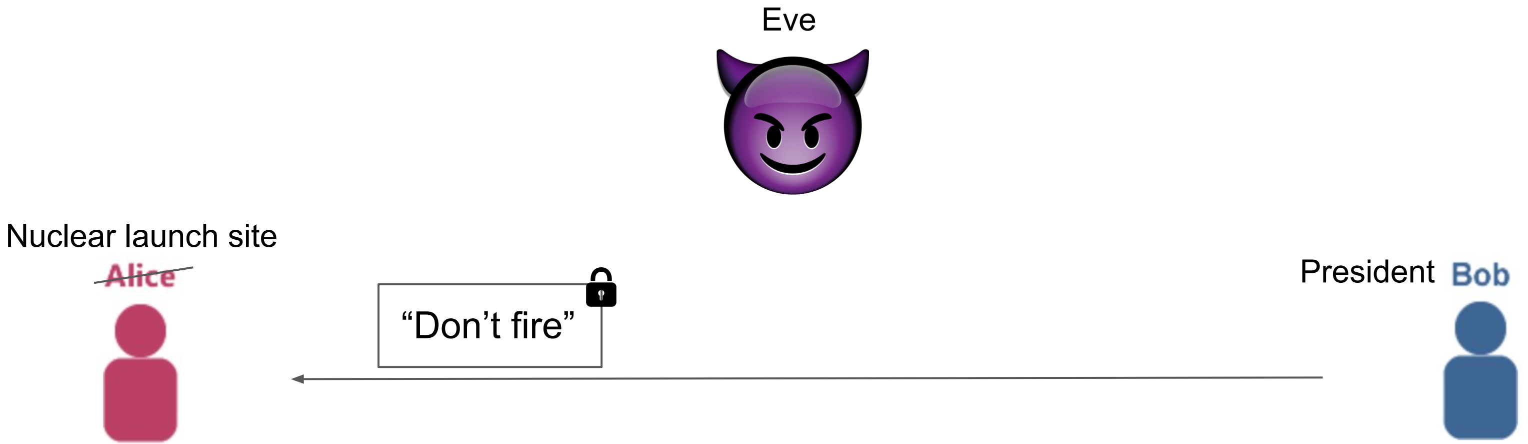
RSA Requires Padding



RSA Requires Padding



RSA Requires Padding



Eve

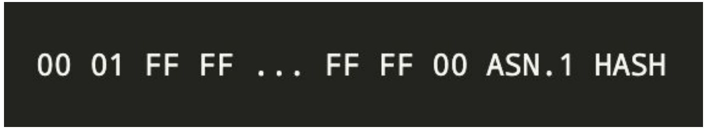
Nuclear launch site

~~Alice~~

“Don’t fire”

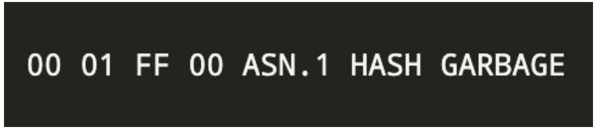
President Bob

Forgery Attack



```
00 01 FF FF ... FF FF 00 ASN.1 HASH
```

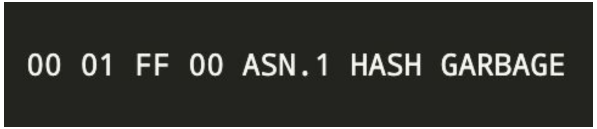
Forgery Attack



```
00 01 FF 00 ASN.1 HASH GARBAGE
```

Forgery Attack

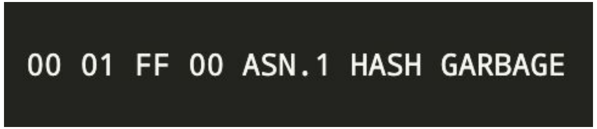
If e = 3, can forge signatures



```
00 01 FF 00 ASN.1 HASH GARBAGE
```

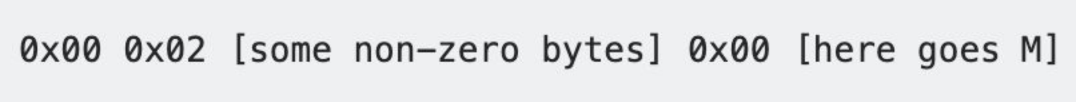
Forgery Attack

If e = 3, can forge signatures



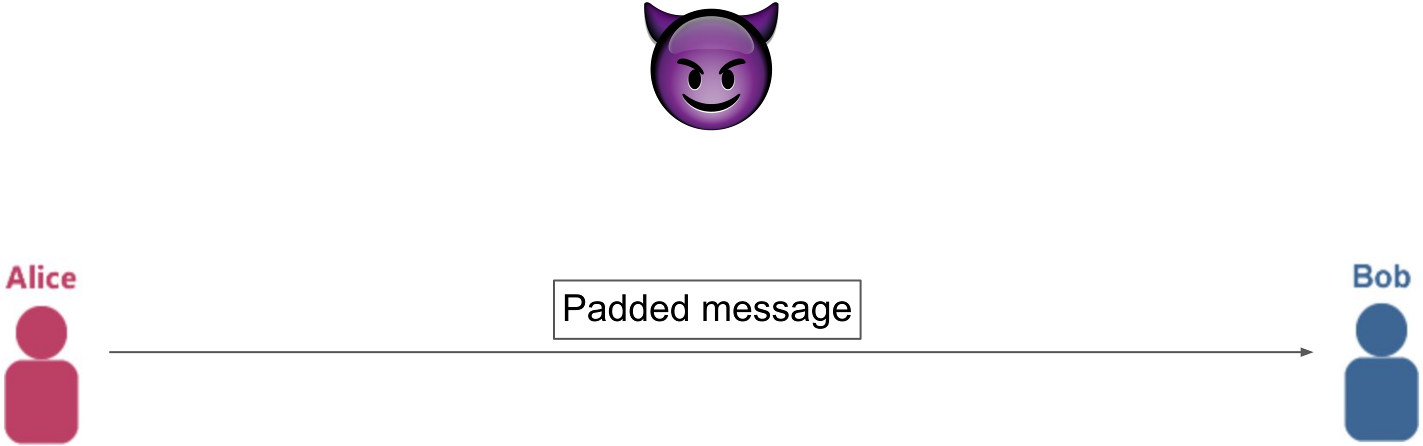
```
00 01 FF 00 ASN.1 HASH GARBAGE
```

Padding Oracle Attacks

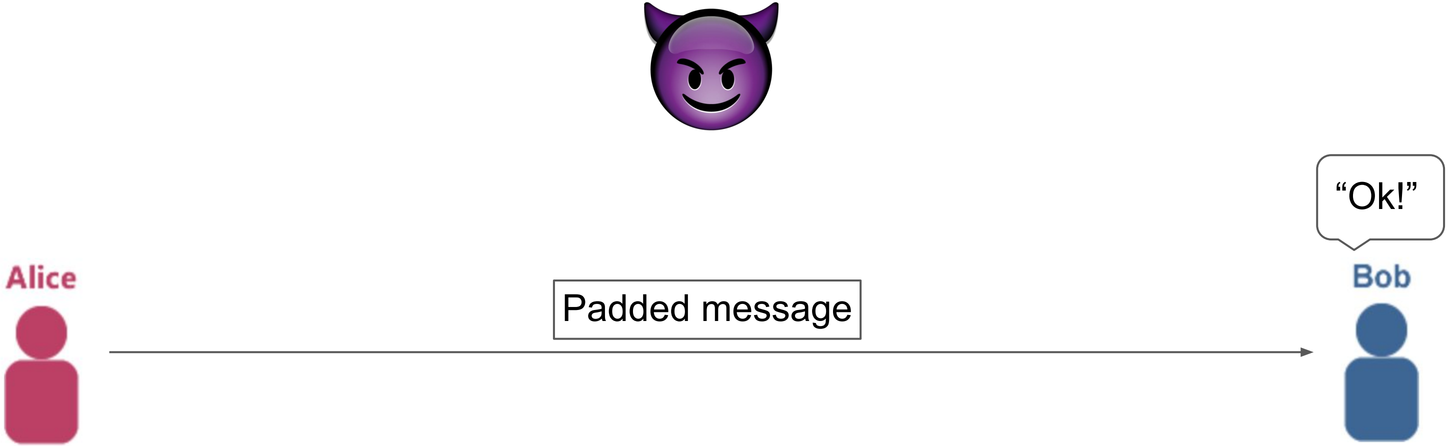


```
0x00 0x02 [some non-zero bytes] 0x00 [here goes M]
```

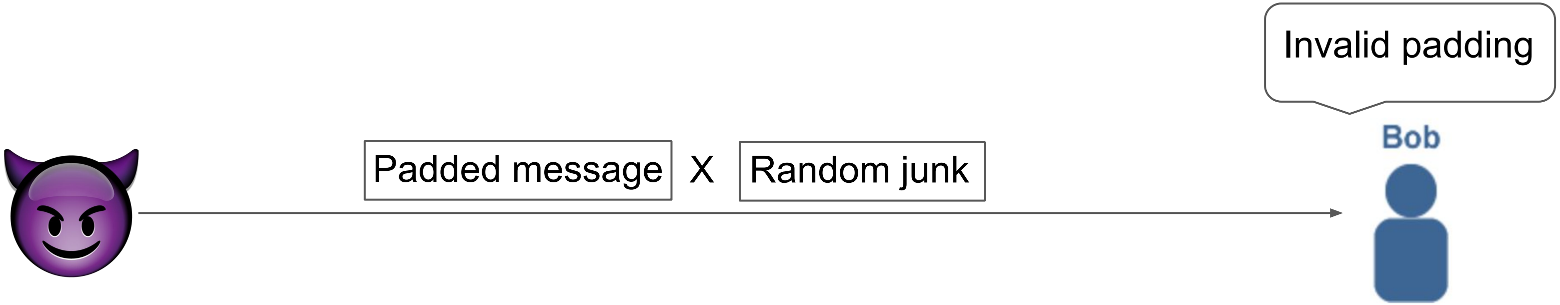
Padding Oracle Attacks



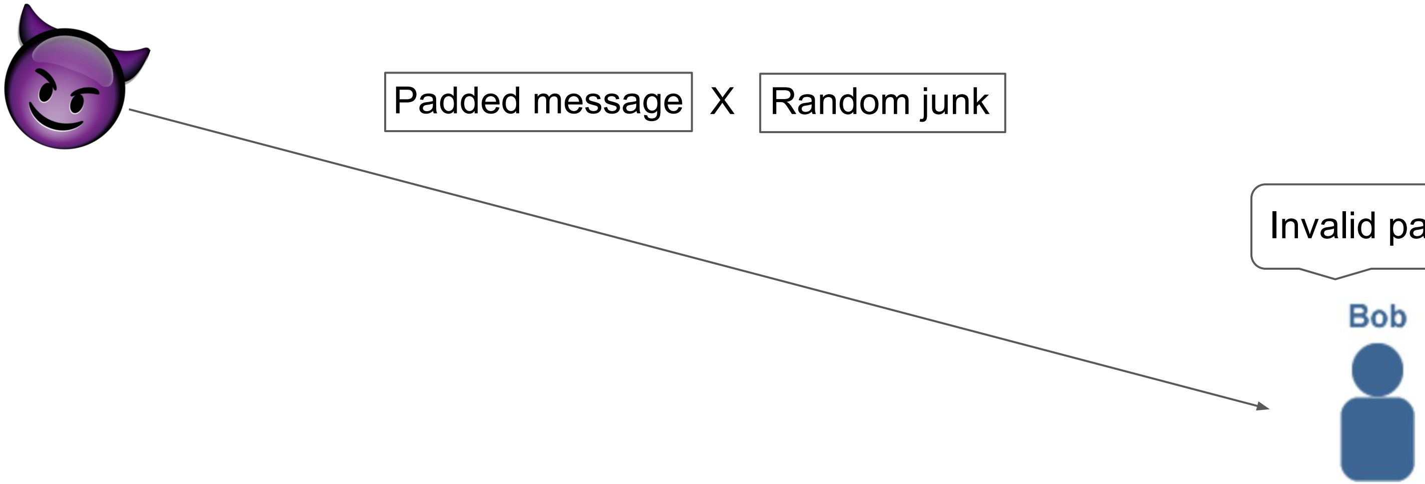
Padding Oracle Attacks



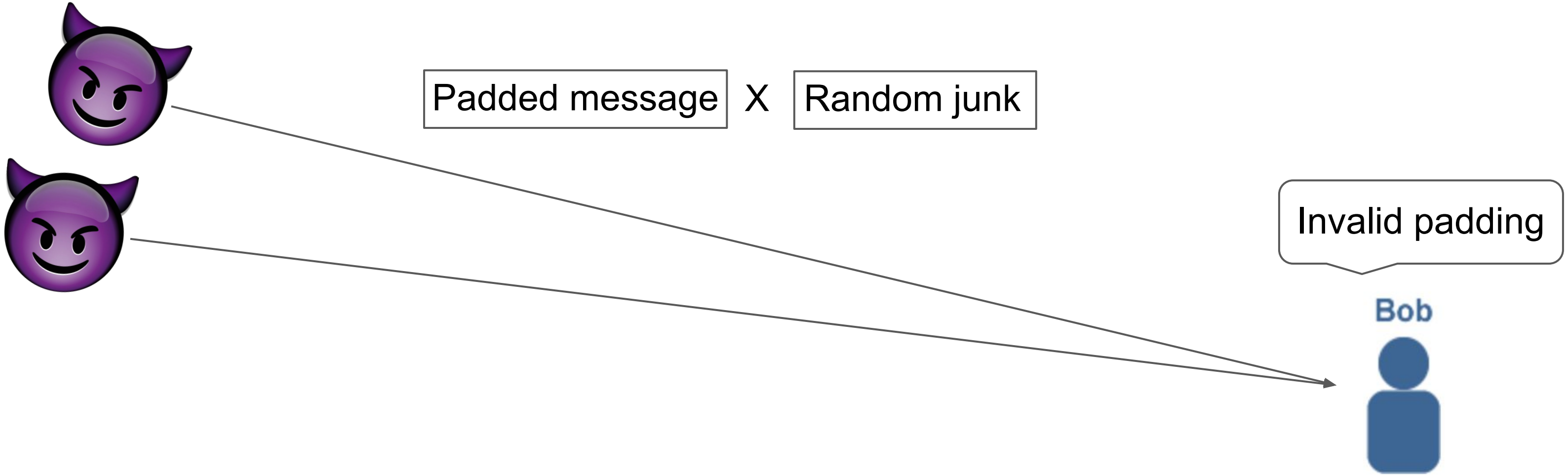
Padding Oracle Attacks



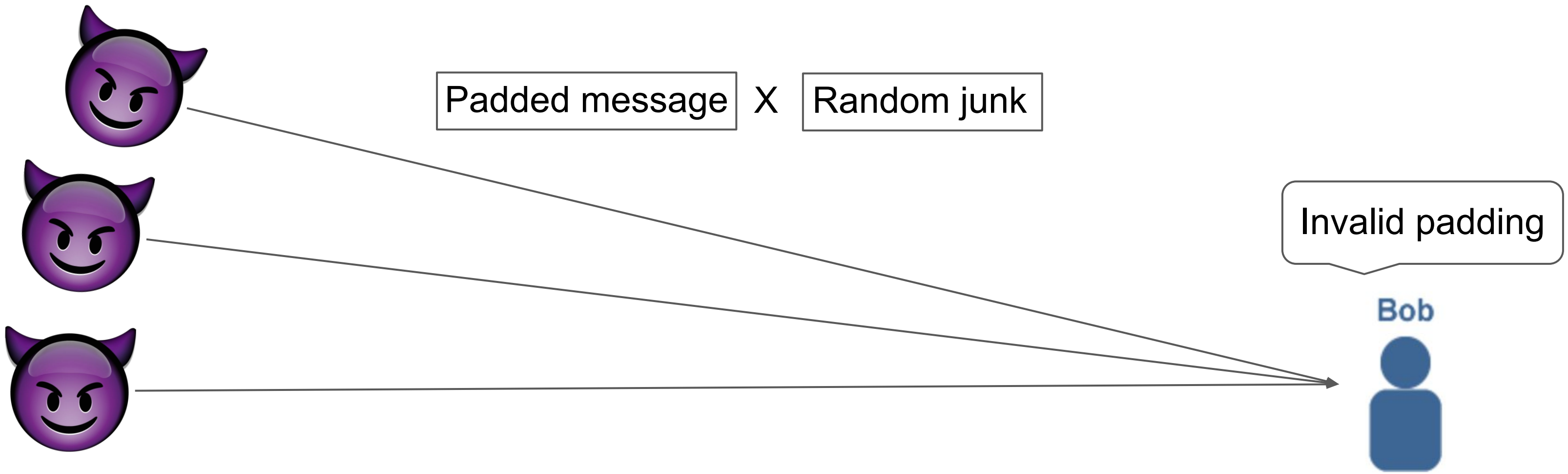
Padding Oracle Attacks



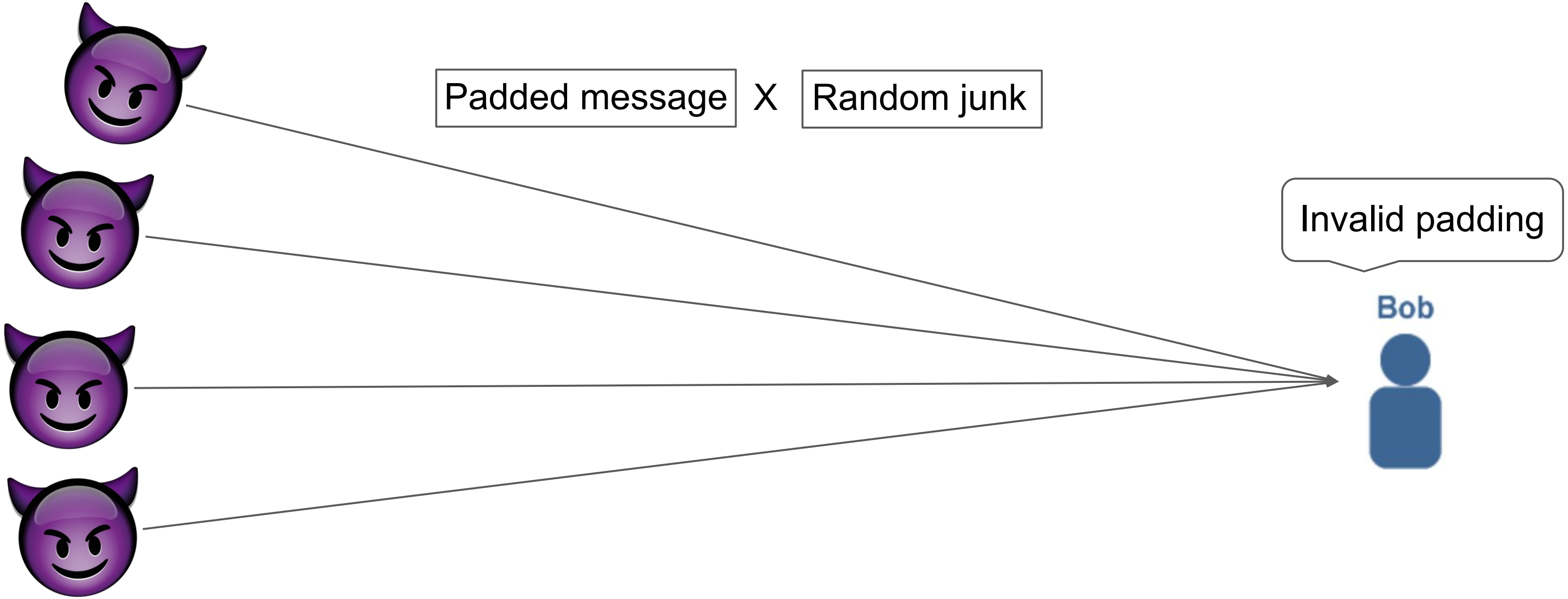
Padding Oracle Attacks



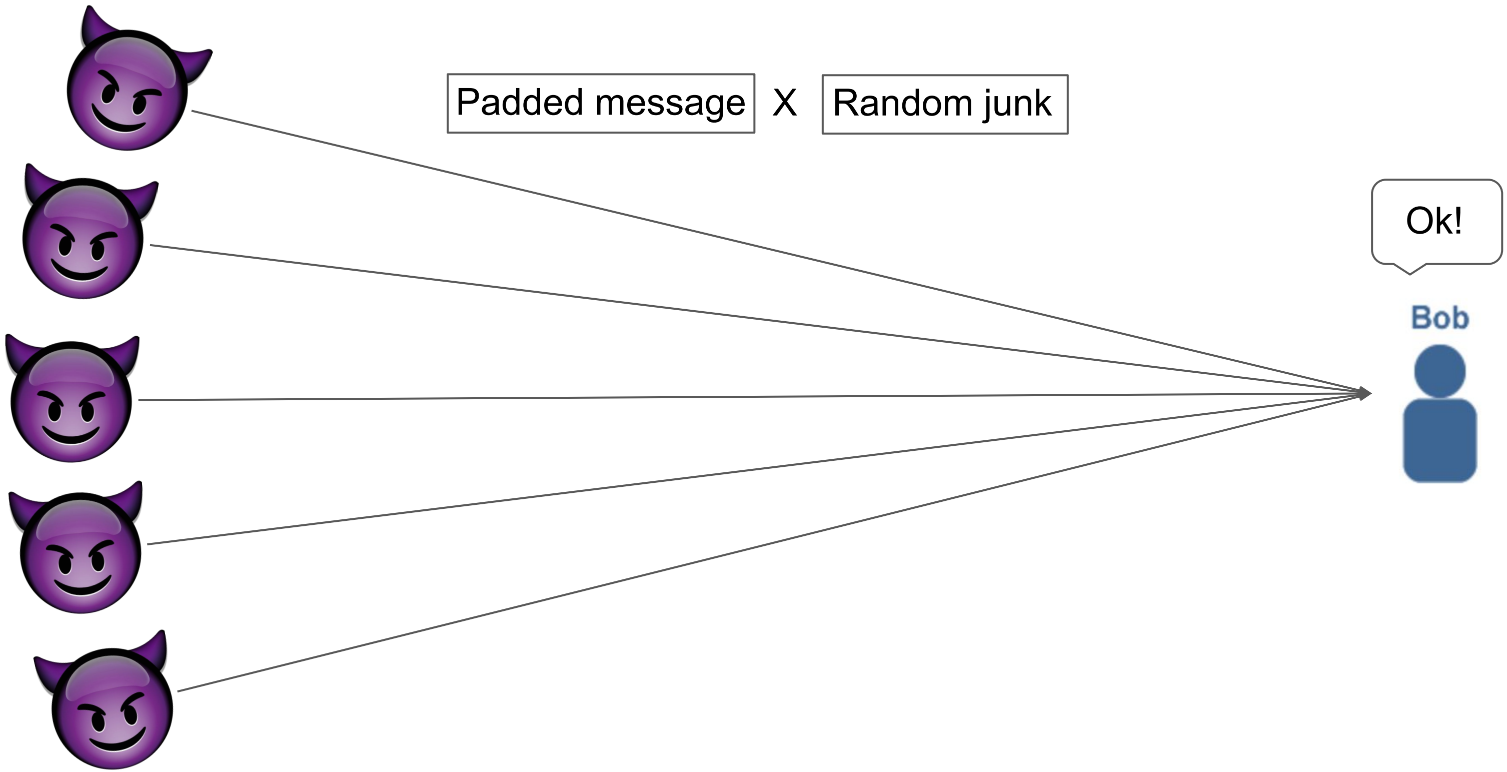
Padding Oracle Attacks



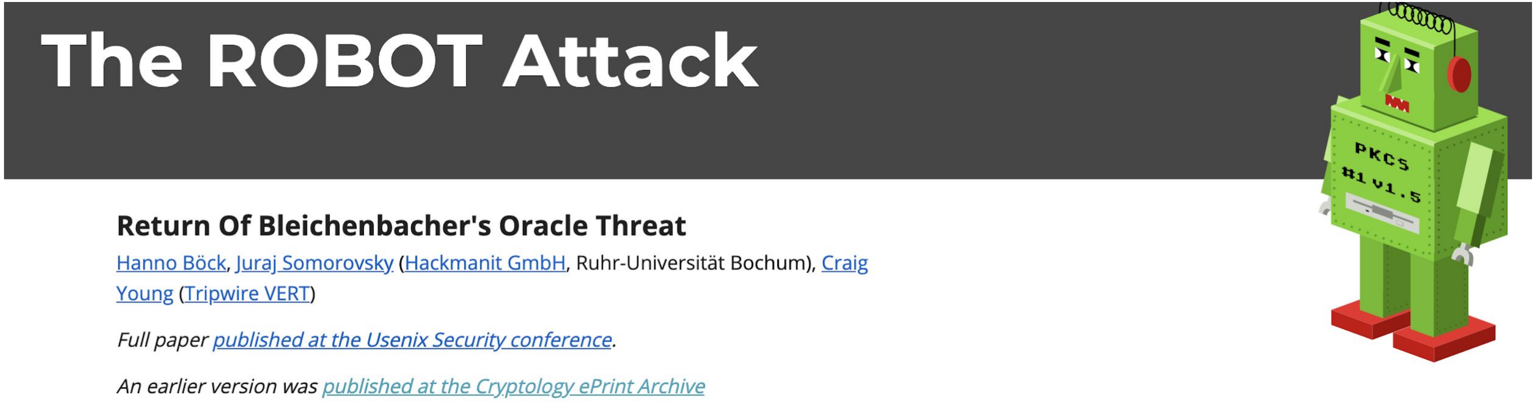
Padding Oracle Attacks



Padding Oracle Attacks



How Bad is This IRL?



The ROBOT Attack

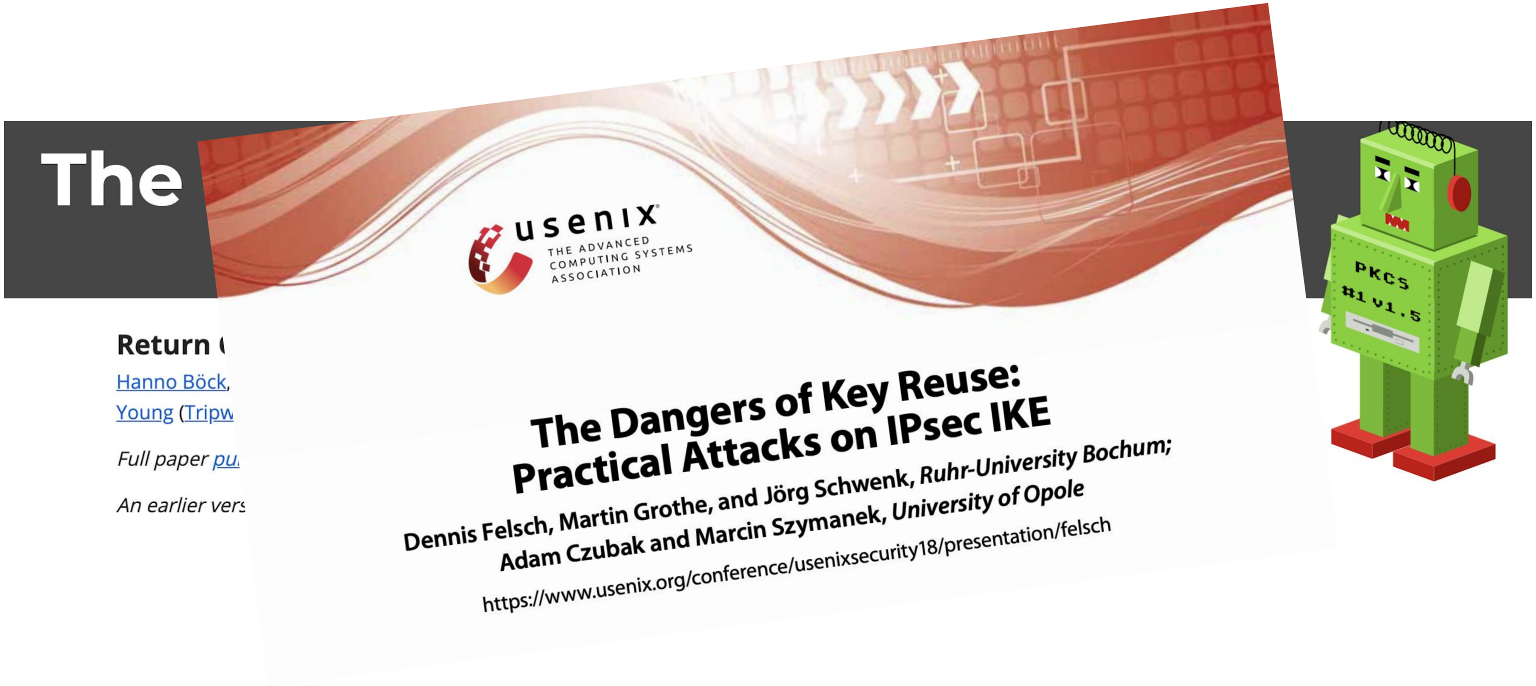
Return Of Bleichenbacher's Oracle Threat

Hanno Böck, Juraj Somorovsky (Hackmanit GmbH, Ruhr-Universität Bochum), Craig Young (Tripwire VERT)

Full paper published at the Usenix Security conference.

An earlier version was published at the Cryptology ePrint Archive

How Bad is This IRL?



The

Return (

Hanno Böck,
Young (Tripw

Full paper pu

An earlier vers

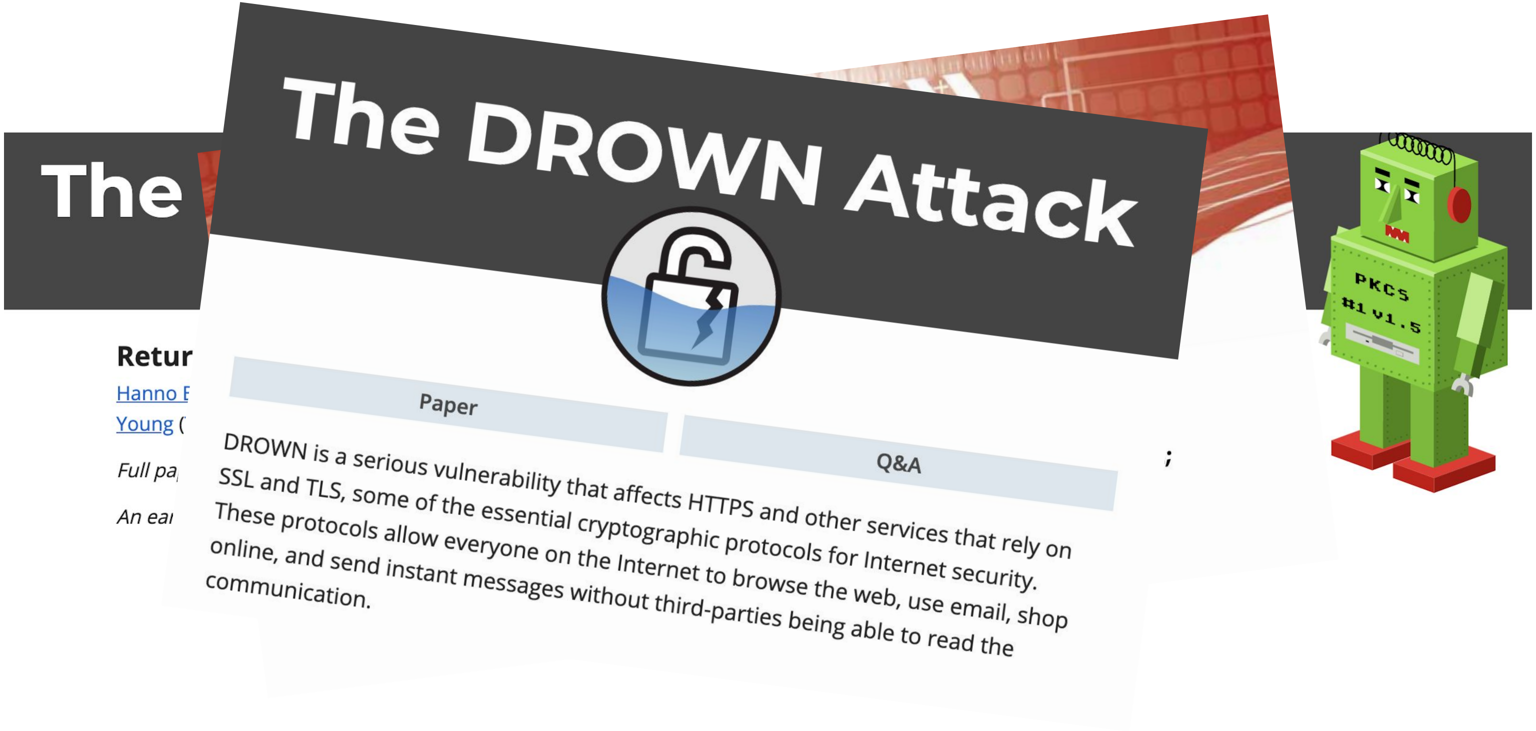
usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

The Dangers of Key Reuse: Practical Attacks on IPsec IKE

Dennis Felsch, Martin Grothe, and Jörg Schwenk, *Ruhr-University Bochum;* Adam Czubak and Marcin Szymanek, *University of Opole*

https://www.usenix.org/conference/usenixsecurity18/presentation/felsch

How Bad is This IRL?



The

Retur

Hanno E

Young (

Full pa

An ea

;

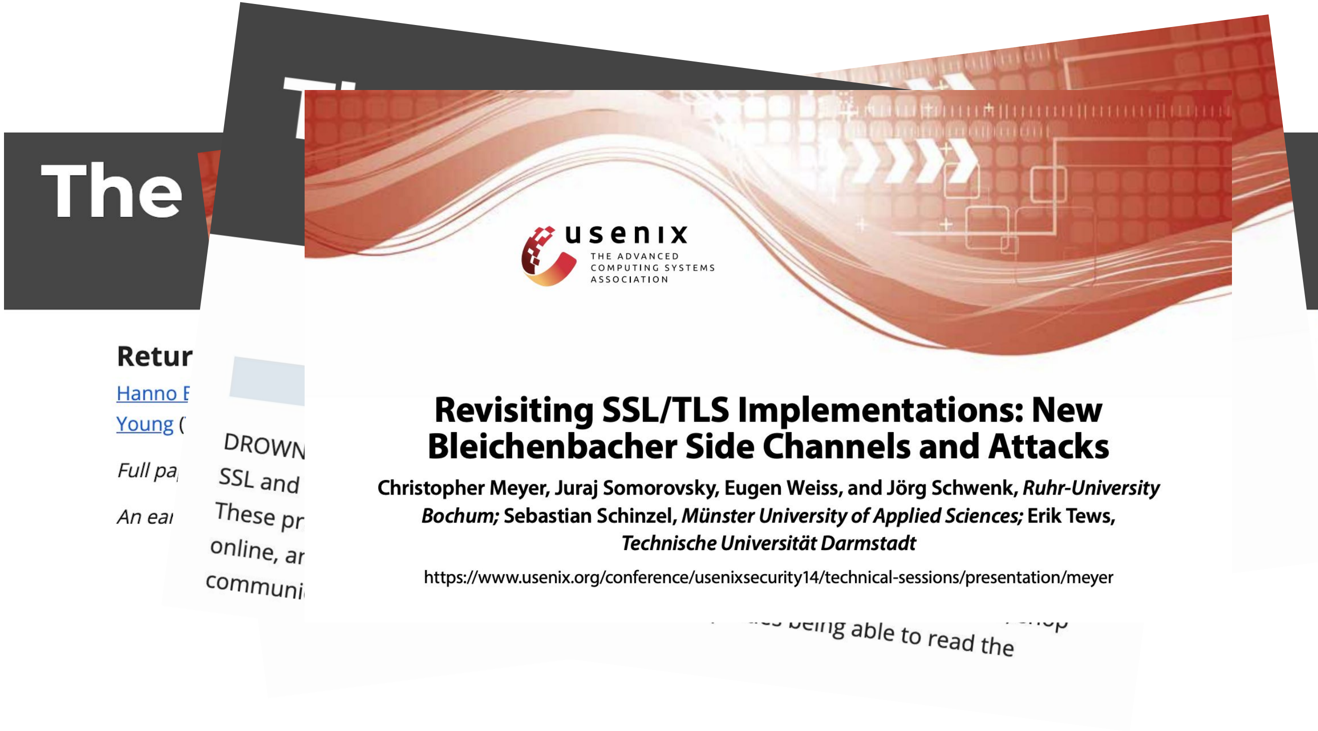
The DROWN Attack

Paper

Q&A

DROWN is a serious vulnerability that affects HTTPS and other services that rely on SSL and TLS, some of the essential cryptographic protocols for Internet security. These protocols allow everyone on the Internet to browse the web, use email, shop online, and send instant messages without third-parties being able to read the communication.

How Bad is This IRL?



The

Retur

Hanno B

Young (

Full pa

An ea

DROWN

SSL and

These pr

online, ar

communi

being able to read the

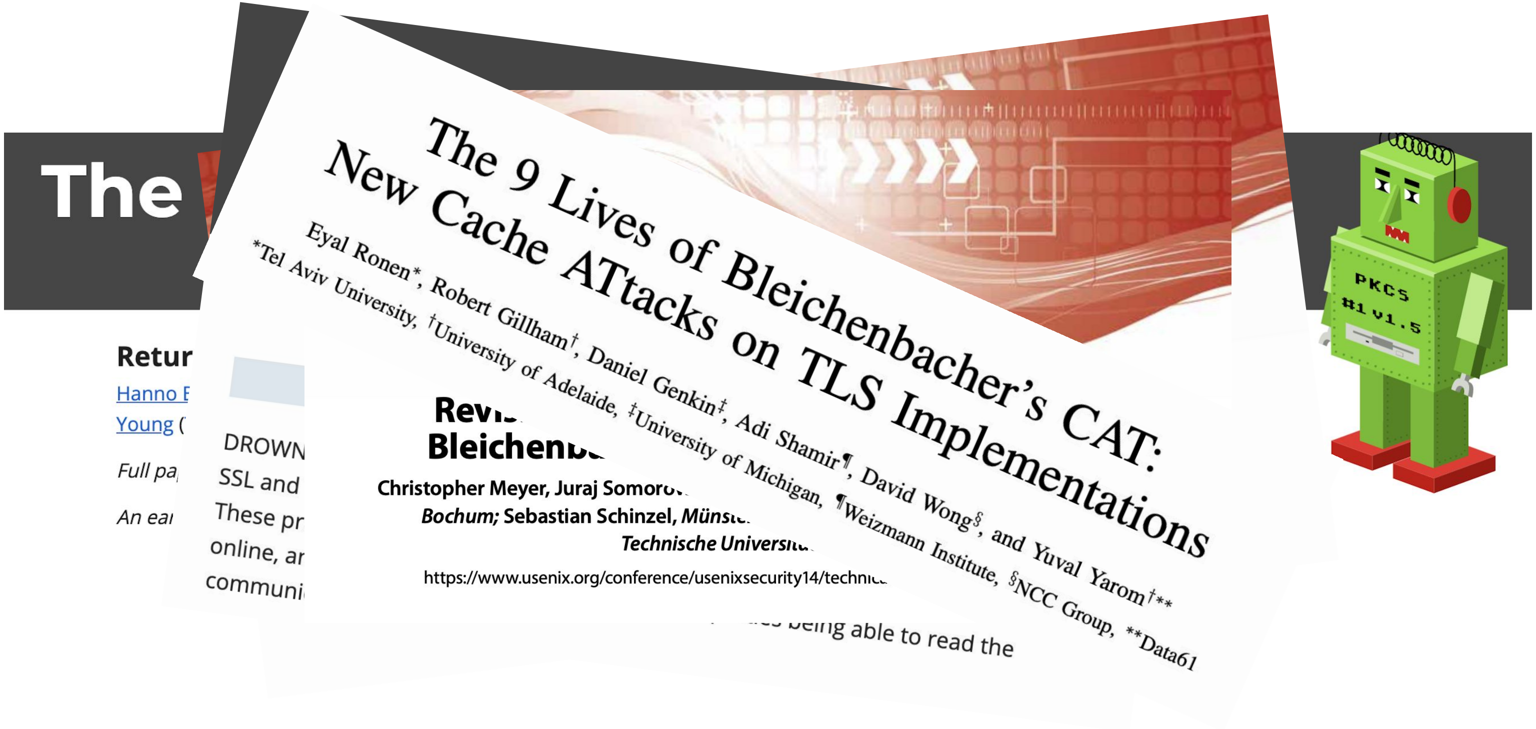
usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Revisiting SSL/TLS Implementations: New Bleichenbacher Side Channels and Attacks

Christopher Meyer, Juraj Somorovsky, Eugen Weiss, and Jörg Schwenk, *Ruhr-University Bochum;* Sebastian Schinzel, *Münster University of Applied Sciences;* Erik Tews, *Technische Universität Darmstadt*

https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/meyer

How Bad is This IRL?



The

Retur

Hanno E

Young (

Full pa

An ear

DROWN

SSL and

These pr

online, ar

communi

Revi

Bleichenb

Christopher Meyer, Juraj Somoro

Bochum; Sebastian Schinzel, Münst

Technische Universit

https://www.usenix.org/conference/usenixsecurity14/technic

being able to read the

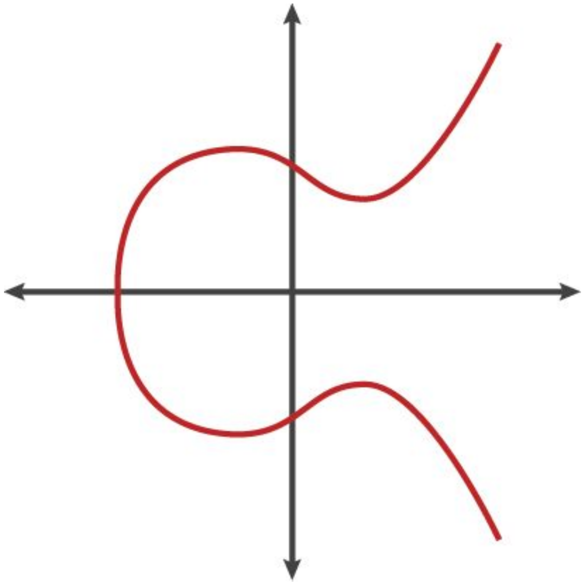
The 9 Lives of Bleichenbacher's CAT:
New Cache ATtacks on TLS Implementations

Eyal Ronen*, Robert Gillham†, Daniel Genkin‡, Adi Shamir¶, David Wong§, and Yuval Yarom†**

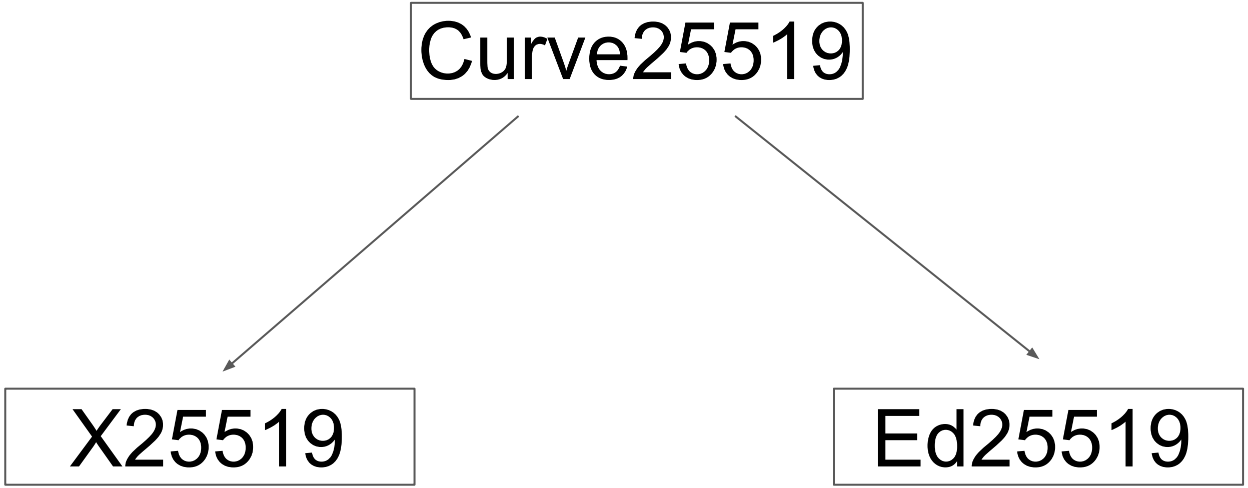
*Tel Aviv University, †University of Adelaide, ‡University of Michigan, ¶Weizmann Institute, §NCC Group, **Data61

What Should I Use Instead?

What Should I Use Instead



What Should I Use Instead?



Curve25519

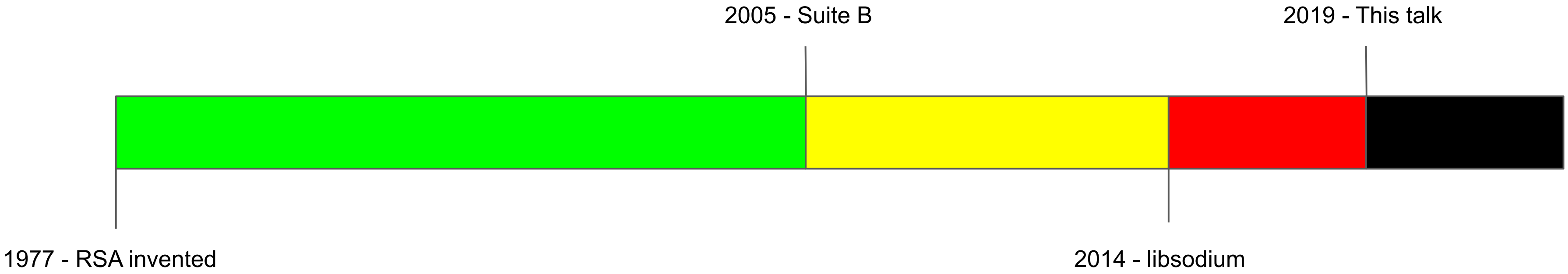
X25519

Ed25519

What Should I Use Instead?

Final Thoughts

RSA Timeline



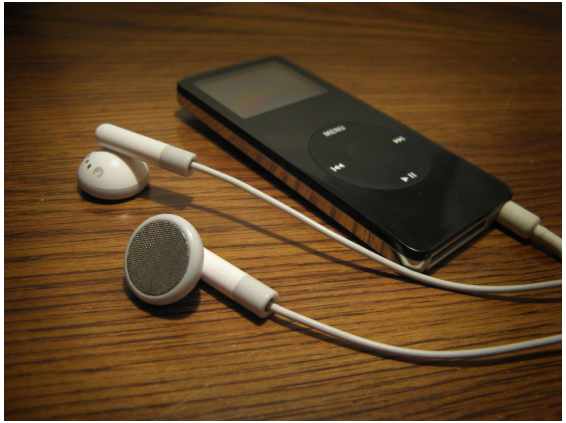
2005 - Suite B

2019 - This talk

1977 - RSA invented

2014 - libsodium

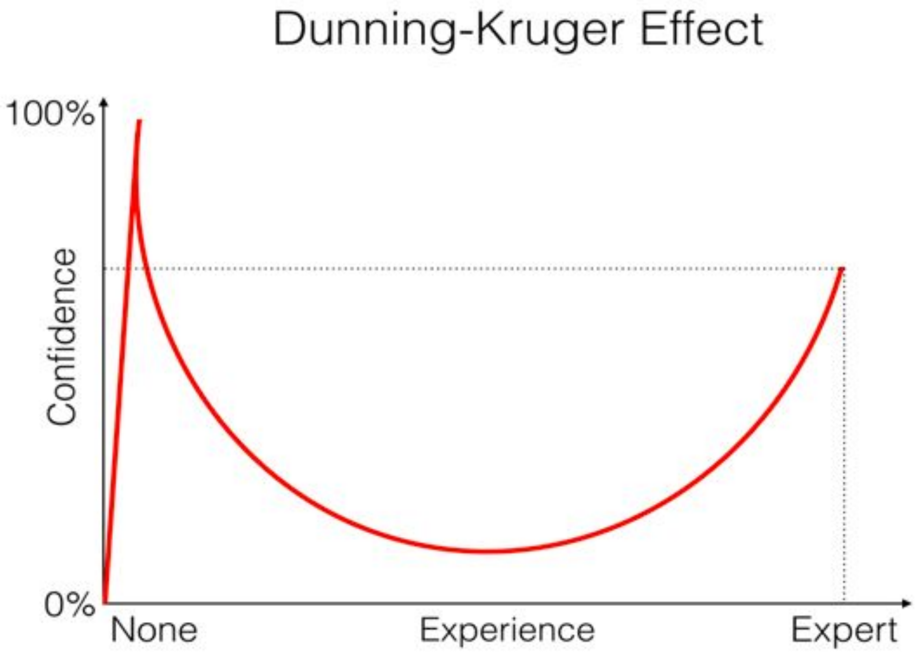
RSA Timeline



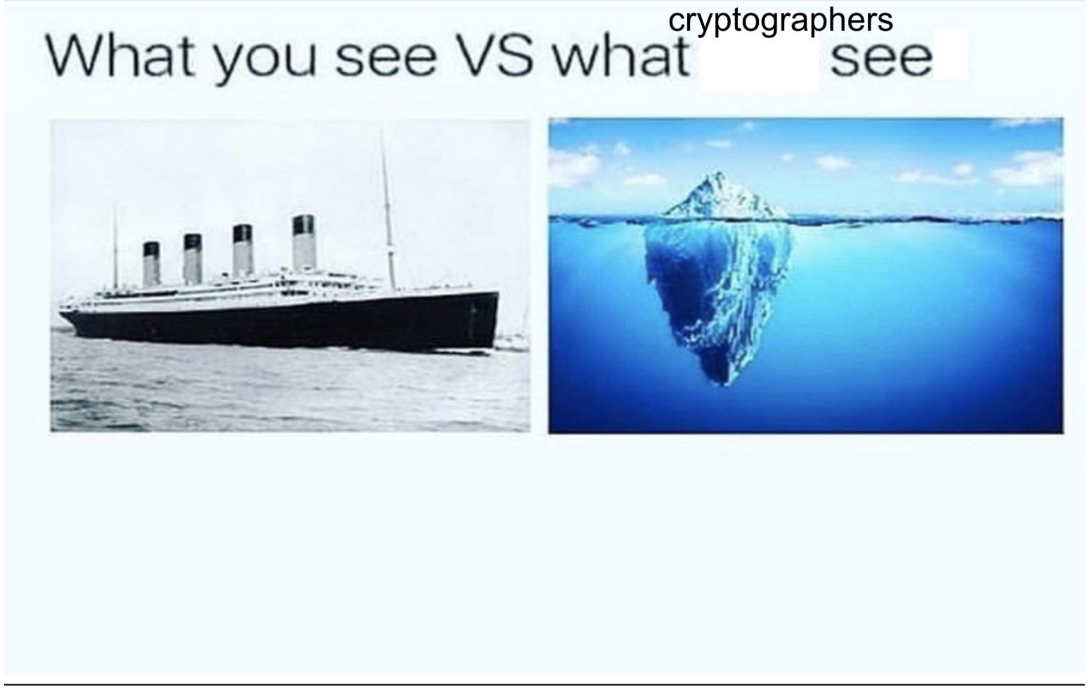
Wrapping Up

Devs talking about their custom RSA implementation
Their RSA implementation

Wrapping Up


Dunning-Kruger Effect
100%
Confidence
0%
None
Experience
Expert

Wrapping Up



Wrapping Up

Wrapping Up

Imagine trying to use digital signatures

this meme was made by

ecc gang

but anyone can sign things as you because you used pkcs#1v1.5 padding wrong

Wrapping Up

“Using crypto in your application shouldn't have to feel like juggling chainsaws in the dark.” - Tink Documentation

Thanks!

Contact

Ben Perez

Security Engineer

benjamin.perez@trailofbits.com

@blperez_