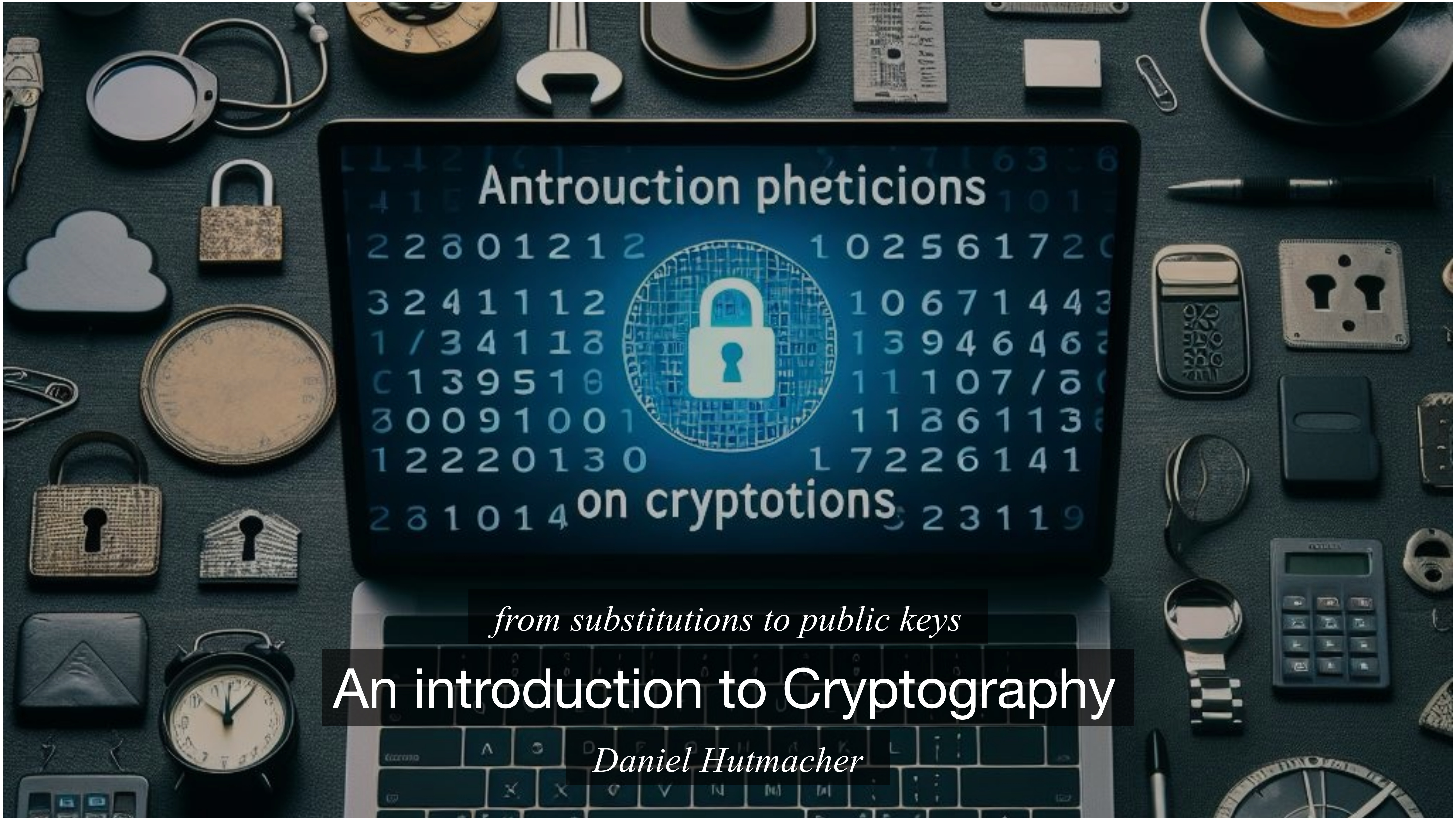


from substitutions to public keys

An introduction to Cryptography

Daniel Hutmacher

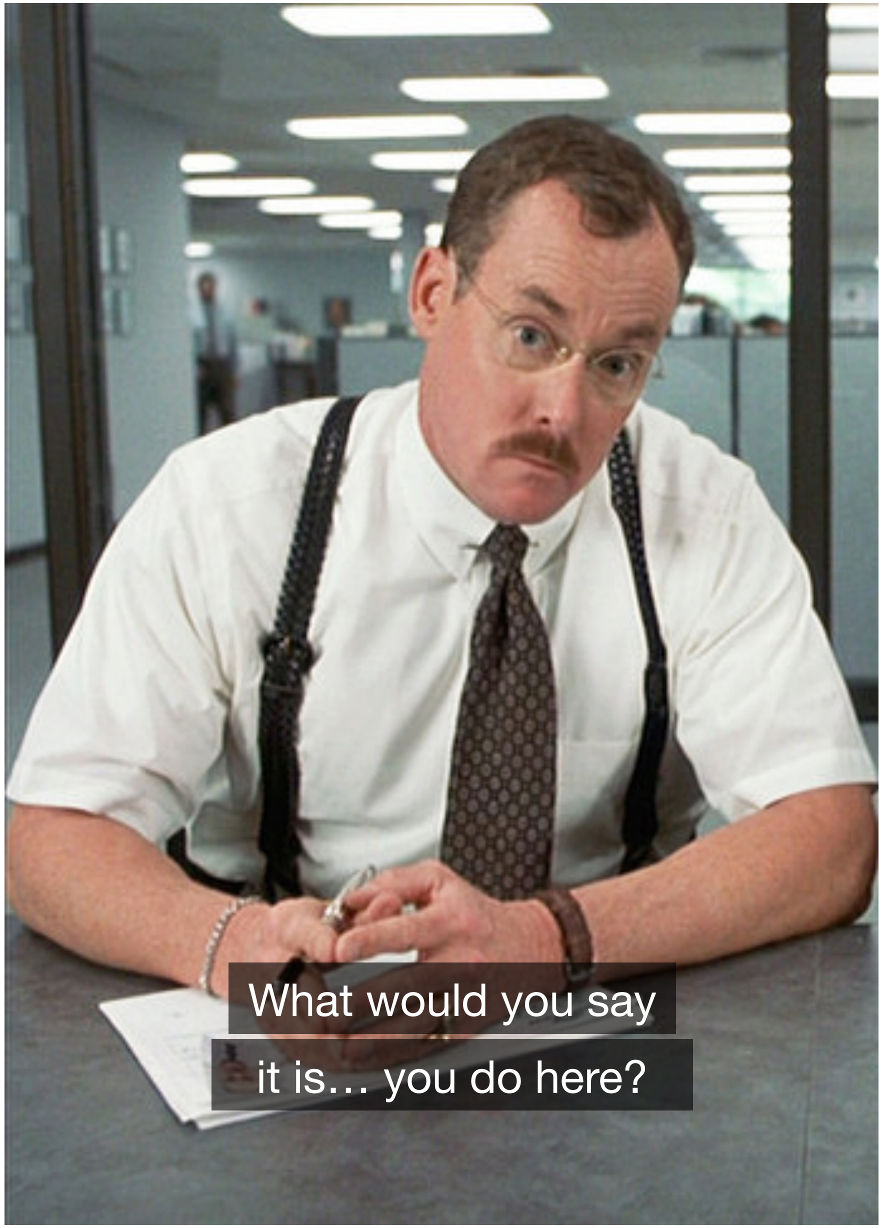

This presentation
Cryptography is pretty complex

Daniel Hutmacher

- SQL Server developer since 1999
- Actually a Data Platform MVP
- Consultant in my own business
- Organizer of Data Saturday Stockholm
- I run dataplatform.social and callfordataspeakers.com

Email: daniel@strd.co
Blog: sqlsunday.com
Bluesky: @dhma.ch

MVP
Microsoft® Most Valuable Professional



Thank you, sponsors!

Caesar shift

Plaintext:	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
Key:	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

Plaintext:	w	e		a	t	t	a	c	k		a	t		d	a	w	n
Ciphertext:	x	f		b	u	u	b	d	l		b	u		e	b	x	o

Substitution

Plaintext:	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
Key:	y	r	q	p	o	t	i	h	x	e	z	j	d	c	b	w	n	m	v	l	k	u	s	g	f	a

Plaintext:	w	e		a	t	t	a	c	k		a	t		d	a	w	n
Ciphertext:	s	o		y	l	l	y	q	z		y	l		p	y	s	c

Substitution ► Frequency analysis

Data source: https://en.wikipedia.org/wiki/Letter_frequency

	English	French	German	Spanish	Portuguese	Italian	Turkish	Swedish	Polish	Dutch	Danish	Icelandic	Finnish	Czech	Hungarian
e	12,7%	14,7%	16,4%	12,2%	12,6%	11,8%	8,9%	10,1%	7,9%	18,9%	15,5%	6,4%	8,0%	7,6%	11,6%
a	8,2%	7,6%	6,5%	11,5%	14,6%	11,7%	11,9%	9,4%	9,0%	7,5%	6,0%	10,1%	12,2%	8,4%	8,9%
n	6,7%	7,1%	9,8%	6,7%	4,4%	6,9%	7,5%	8,5%	5,6%	10,0%	7,2%	7,7%	8,8%	6,5%	6,8%
i	7,0%	7,5%	6,6%	6,2%	6,2%	10,1%	8,6%	5,8%	8,3%	6,5%	6,0%	7,6%	10,8%	6,1%	4,3%
r	6,0%	6,7%	7,0%	6,9%	6,5%	6,4%	6,7%	8,4%	4,6%	6,4%	9,0%	8,6%	2,9%	4,8%	2,7%
t	9,1%	7,2%	6,2%	4,6%	4,3%	5,6%	3,3%	7,7%	4,0%	6,8%	6,9%	5,0%	8,8%	5,7%	7,0%
s	6,3%	7,9%	7,3%	8,0%	6,8%	5,0%	3,0%	6,6%	4,3%	3,7%	5,8%	5,6%	7,9%	5,2%	7,0%
o	7,5%	5,8%	2,6%	8,7%	9,7%	9,8%	2,5%	4,5%	7,6%	6,1%	4,6%	2,2%	5,6%	6,7%	3,7%
l	4,0%	5,5%	3,4%	5,0%	2,8%	6,5%	5,9%	5,3%	2,1%	3,6%	5,2%	4,5%	5,8%	3,8%	6,7%
d	4,3%	3,7%	5,1%	5,0%	5,0%	3,7%	4,7%	4,7%	3,3%	5,9%	5,9%	1,6%	1,0%	3,5%	1,9%
m	2,4%	3,0%	2,5%	3,2%	4,7%	2,5%	3,8%	3,5%	2,9%	2,2%	3,2%	4,0%	3,2%	2,4%	3,8%
u	2,8%	6,3%	4,2%	3,9%	3,6%	2,8%	3,2%	1,9%	2,3%	2,0%	2,0%	4,6%	5,0%	2,2%	0,4%
k	1,8%	0,1%	1,4%	0,0%	0,0%	0,0%	4,7%	3,1%	3,4%	2,3%	3,4%	3,3%	5,0%	2,9%	4,9%
g	2,0%	0,9%	3,0%	1,8%	1,3%	1,6%	1,3%	2,9%	1,4%	3,4%	4,1%	4,2%	0,4%	0,1%	3,8%
c	2,8%	3,3%	2,7%	4,0%	3,9%	4,5%	1,0%	1,5%	4,0%	1,2%	0,6%		0,3%	0,7%	0,6%
h	6,1%	0,9%	4,6%	2,0%	1,3%	0,1%	1,2%	2,1%	1,1%	2,4%	1,6%	1,9%	1,9%	1,4%	1,3%
v	1,0%	1,8%	0,8%	1,1%	1,6%	2,1%	1,0%	2,4%	0,0%	2,9%	2,3%	2,4%	2,3%	5,3%	2,3%
p	1,9%	2,5%	0,7%	2,5%	2,5%	3,1%	0,9%	1,8%	3,1%	1,6%	1,8%	0,8%	1,8%	1,9%	0,5%
b	1,5%	0,9%	1,9%	2,2%	1,0%	0,9%	2,8%	1,5%	1,5%	1,6%	2,0%	1,0%	0,3%	0,8%	1,9%
y	2,0%	0,7%	0,0%	1,4%	0,0%	0,0%	3,3%	0,7%	3,9%	0,0%	0,7%	0,9%	1,7%	1,0%	2,6%
z	0,1%	0,3%	1,1%	0,5%	0,5%	1,2%	1,5%	0,1%	5,6%	1,4%	0,0%		0,1%	1,6%	4,3%
f	2,2%	1,1%	1,7%	0,7%	1,0%	1,2%	0,5%	2,0%	0,3%	0,8%	2,4%	3,0%	0,2%	0,1%	0,5%
j	0,3%	0,8%	0,3%	2,5%	0,9%	0,0%	0,0%	0,6%	2,3%	1,5%	0,7%	1,1%	2,0%	1,4%	1,5%
w	2,4%	0,0%	1,9%	0,0%	0,0%	0,0%		0,1%	4,5%	1,5%	0,1%		0,1%	0,0%	
é		1,5%		0,4%	0,3%							0,6%		0,6%	4,3%
á				0,5%	0,1%							1,8%		0,9%	3,4%
ä			0,6%					1,8%					3,6%		
ı							5,1%								
í				0,7%	0,1%							1,6%		1,6%	0,5%

Frequency analysis

	Ciphertext		English
a	0,5%	a	8,2%
b	1,3%	b	1,5%
c		c	2,8%
d	2,0%	d	4,3%
e	0,9%	e	12,7%
f	2,4%	f	2,2%
g	9,0%	g	2,0%
h	6,2%	h	6,1%
i	4,2%	i	7,0%
j		j	0,3%
k	1,4%	k	1,8%
l	8,0%	l	4,0%
m	6,5%	m	2,4%
n	2,0%	n	6,7%
o	3,1%	o	7,5%
p	1,1%	p	1,9%
q	0,8%	q	
r	7,1%	r	6,0%
s	5,9%	s	6,3%
t	1,8%	t	9,1%
u	1,9%	u	2,8%
v	12,2%	v	1,0%
w	4,0%	w	2,4%
x	2,1%	x	

```
dv'iv ml hgizmtvih gl olev
blf pmld gsv ifovh zmw hl wl r
z ufoo xlnnrgnvmg'h dszg r'n gsrmprmt lu
blf dlfowm'g tvg gsrh uiln zmb lgsvi tfb
r qfhg dzmmz gvoo blf sld r'n uvvormt
tlggz nzpv blf fmwvihgzmw
mvevi tlmmz trev blf fk
mvevi tlmmz ovg blf wldm
mvevi tlmmz ifm zilfmw zmw wvhvig blf
mvevi tlmmz nzpv blf xib
mvevi tlmmz hzb tllwybv
mvevi tlmmz gvoo z orv zmw sfig blf
dv'ev pmldm vzxs lgsvi uli hl olmt
blfi svzig'h yvvm zxsrmt, yfg blf'iv gll hsb gl hzb rg
rmhrwv, dv ylgs pmld dszg'h yvvm tlrmt lm
dv pmld gsv tznv zmw dv'iv tlmmz kozb rg
zmw ru blf zhp nv sld r'n uvvormt
wlm'g gvoo nv blf'iv gll yormw gl hvv
mvevi tlmmz trev blf fk
mvevi tlmmz ovg blf wldm
mvevi tlmmz ifm zilfmw zmw wvhvig blf
mvevi tlmmz nzpv blf xib
mvevi tlmmz hzb tllwybv
mvevi tlmmz gvoo z orv zmw sfig blf
mvevi tlmmz trev blf fk
mvevi tlmmz ovg blf wldm
mvevi tlmmz ifm zilfmw zmw wvhvig blf
mvevi tlmmz nzpv blf xib
mvevi tlmmz hzb tllwybv
mvevi tlmmz gvoo z orv zmw sfig blf
dv'ev pmldm vzxs lgsvi uli hl olmt
blfi svzig'h yvvm zxsrmt, yfg blf'iv gll hsb gl hzb rg
rmhrwv, dv ylgs pmld dszg'h yvvm tlrmt lm
dv pmld gsv tznv zmw dv'iv tlmmz kozb rg
r qfhg dzmmz gvoo blf sld r'n uvvormt
tlggz nzpv blf fmwvihgzmw
```

Frequency analysis

	Ciphertext		English
v	12,2%	e	12,7%
g	9,0%	t	9,1%
l	8,0%	a	8,2%
z	7,9%	o	7,5%
r	7,1%	i	7,0%
m	6,5%	n	6,7%
h	6,2%	s	6,3%
s	5,9%	h	6,1%
i	4,2%	r	6,0%
w	4,0%	d	4,3%
o	3,1%	l	4,0%
f	2,4%	c	2,8%
x	2,1%	u	2,8%
n	2,0%	m	2,4%
d	2,0%	w	2,4%
u	1,9%	f	2,2%
t	1,8%	g	2,0%
k	1,4%	y	2,0%
b	1,3%	p	1,9%
p	1,1%	k	1,8%
y	1,0%	b	1,5%
e	0,9%	v	1,0%
q	0,8%	j	0,3%
a	0,5%	z	0,1%

```
dv'iv ml hgizmtvih gl olev
blf pmld gsv ifovh zmw hl wl r
z ufoo xlnnrgnvmg'h dszg r'n gsrmprmt lu
blf dlfowm'g tvg gsrh uiln zmb lgsvi tfb
r qfhg dzmmz gvoo blf sld r'n uvvormt
tlggz nzpv blf fmwvihgzmw
mvevi tlmmz trev blf fk
mvevi tlmmz ovg blf wldm
mvevi tlmmz ifm zilfmw zmw wvhvig blf
mvevi tlmmz nzpv blf xib
mvevi tlmmz hzb tllwybv
mvevi tlmmz gvoo z orv zmw sfig blf
dv'ev pmldm vzxs lgsvi uli hl olmt
blfi svzig'h yvvm zxsrmt, yfg blf'iv gll hsb gl hzb rg
rmhrwv, dv ylgs pmld dszg'h yvvm tlrmt lm
dv pmld gsv tznv zmw dv'iv tlmmz kozb rg
zmw ru blf zhp nv sld r'n uvvormt
wlm'g gvoo nv blf'iv gll yormw gl hvv
mvevi tlmmz trev blf fk
mvevi tlmmz ovg blf wldm
mvevi tlmmz ifm zilfmw zmw wvhvig blf
mvevi tlmmz nzpv blf xib
mvevi tlmmz hzb tllwybv
mvevi tlmmz gvoo z orv zmw sfig blf
mvevi tlmmz trev blf fk
mvevi tlmmz ovg blf wldm
mvevi tlmmz ifm zilfmw zmw wvhvig blf
mvevi tlmmz nzpv blf xib
mvevi tlmmz hzb tllwybv
mvevi tlmmz gvoo z orv zmw sfig blf
dv'ev pmldm vzxs lgsvi uli hl olmt
blfi svzig'h yvvm zxsrmt, yfg blf'iv gll hsb gl hzb rg
rmhrwv, dv ylgs pmld dszg'h yvvm tlrmt lm
dv pmld gsv tznv zmw dv'iv tlmmz kozb rg
r qfhg dzmmz gvoo blf sld r'n uvvormt
tlggz nzpv blf fmwvihgzmw
```

Frequency analysis

	Ciphertext		English
v	12,2%	e	12,7%
g	9,0%	t	9,1%
l	8,0%	a	8,2%
z	7,9%	o	7,5%
r	7,1%	i	7,0%
m	6,5%	n	6,7%
h	6,2%	s	6,3%
s	5,9%	h	6,1%
i	4,2%	r	6,0%
w	4,0%	d	4,3%
o	3,1%	l	4,0%
f	2,4%	c	2,8%
x	2,1%	u	2,8%
n	2,0%	m	2,4%
d	2,0%	w	2,4%
u	1,9%	f	2,2%
t	1,8%	g	2,0%
k	1,4%	y	2,0%
b	1,3%	p	1,9%
p	1,1%	k	1,8%
y	1,0%	b	1,5%
e	0,9%	v	1,0%
q	0,8%	j	0,3%
a	0,5%	z	0,1%

```
we're na strongers ta lave
pac knaw the rcles ond sa da i
o fcll uammitment's whot i'm thinking af
pac wacldn't get this fram onp ather gcp
i jcst wonno tell pac haw i'm feeling
gatto moke pac cnderstond
never ganno give pac cy
never ganno let pac dawn
never ganno rcn oracnd ond desert pac
never ganno moke pac urp
never ganno sop gaadbpe
never ganno tell o lie ond hcrt pac
we've knawn eouh ather far sa lang
pacr heort's been ouhing, bct pac're taa shp ta sop it
inside, we bath knaw whot's been gaing an
we knaw the gome ond we're ganno ylop it
ond if pac osk me haw i'm feeling
dan't tell me pac're taa blind ta see
never ganno give pac cy
never ganno let pac dawn
never ganno rcn oracnd ond desert pac
never ganno moke pac urp
never ganno sop gaadbpe
never ganno tell o lie ond hcrt pac
never ganno give pac cy
never ganno let pac dawn
never ganno rcn oracnd ond desert pac
never ganno moke pac urp
never ganno sop gaadbpe
never ganno tell o lie ond hcrt pac
we've knawn eouh ather far sa lang
pacr heort's been ouhing, bct pac're taa shp ta sop it
inside, we bath knaw whot's been gaing an
we knaw the gome ond we're ganno ylop it
i jcst wonno tell pac haw i'm feeling
gatto moke pac cnderstond
```

Frequency analysis

	Ciphertext		English
v	12,2%	e	12,7%
g	9,0%	t	9,1%
l	8,0%	a	8,2%
z	7,9%	o	7,5%
r	7,1%	i	7,0%
m	6,5%	n	6,7%
h	6,2%	s	6,3%
s	5,9%	h	6,1%
i	4,2%	r	6,0%
w	4,0%	d	4,3%
o	3,1%	l	4,0%
f	2,4%	c	2,8%
x	2,1%	u	2,8%
n	2,0%	m	2,4%
d	2,0%	w	2,4%
u	1,9%	f	2,2%
t	1,8%	g	2,0%
k	1,4%	y	2,0%
b	1,3%	p	1,9%
p	1,1%	k	1,8%
y	1,0%	b	1,5%
e	0,9%	v	1,0%
q	0,8%	j	0,3%
a	0,5%	z	0,1%

```
we're no strangers to love
poc know the rcles and so do i
a fcll uommitment's what i'm thinking of
poc wocldn't get this from anp other gcp
i jcst wanna tell poc how i'm feeling
gotta make poc cnderstand
never gonna give poc cy
never gonna let poc down
never gonna rcn arocnd and desert poc
never gonna make poc urp
never gonna sap goodbpe
never gonna tell a lie and hcrt poc
we've known eauh other for so long
pocr heart's been auhing, bct poc're too shp to sap it
inside, we both know what's been going on
we know the game and we're gonna ylap it
and if poc ask me how i'm feeling
don't tell me poc're too blind to see
never gonna give poc cy
never gonna let poc down
never gonna rcn arocnd and desert poc
never gonna make poc urp
never gonna sap goodbpe
never gonna tell a lie and hcrt poc
never gonna give poc cy
never gonna let poc down
never gonna rcn arocnd and desert poc
never gonna make poc urp
never gonna sap goodbpe
never gonna tell a lie and hcrt poc
we've known eauh other for so long
pocr heart's been auhing, bct poc're too shp to sap it
inside, we both know what's been going on
we know the game and we're gonna ylap it
i jcst wanna tell poc how i'm feeling
gotta make poc cnderstand
```

Frequency analysis

	Ciphertext		English
v	12,2%	e	12,7%
g	9,0%	t	9,1%
l	8,0%	a	8,2%
z	7,9%	o	7,5%
r	7,1%	i	7,0%
m	6,5%	n	6,7%
h	6,2%	s	6,3%
s	5,9%	h	6,1%
i	4,2%	r	6,0%
w	4,0%	d	4,3%
o	3,1%	l	4,0%
f	2,4%	c	2,8%
x	2,1%	u	2,8%
n	2,0%	m	2,4%
d	2,0%	w	2,4%
u	1,9%	f	2,2%
t	1,8%	g	2,0%
k	1,4%	y	2,0%
b	1,3%	p	1,9%
p	1,1%	k	1,8%
y	1,0%	b	1,5%
e	0,9%	v	1,0%
q	0,8%	j	0,3%
a	0,5%	z	0,1%

```
we're no strangers to love
pou know the rules and so do i
a full commitment's what i'm thinking of
pou wouldn't get this from anp other gup
i just wanna tell pou how i'm feeling
gotta make pou understand
never gonna give pou uy
never gonna let pou down
never gonna run around and desert pou
never gonna make pou crp
never gonna sap goodbpe
never gonna tell a lie and hurt pou
we've known each other for so long
pour heart's been aching, but pou're too shp to sap it
inside, we both know what's been going on
we know the game and we're gonna ylap it
and if pou ask me how i'm feeling
don't tell me pou're too blind to see
never gonna give pou uy
never gonna let pou down
never gonna run around and desert pou
never gonna make pou crp
never gonna sap goodbpe
never gonna tell a lie and hurt pou
never gonna give pou uy
never gonna let pou down
never gonna run around and desert pou
never gonna make pou crp
never gonna sap goodbpe
never gonna tell a lie and hurt pou
we've known each other for so long
pour heart's been aching, but pou're too shp to sap it
inside, we both know what's been going on
we know the game and we're gonna ylap it
i just wanna tell pou how i'm feeling
gotta make pou understand
```

Frequency analysis

	Ciphertext		English
v	12,2%	e	12,7%
g	9,0%	t	9,1%
l	8,0%	a	8,2%
z	7,9%	o	7,5%
r	7,1%	i	7,0%
m	6,5%	n	6,7%
h	6,2%	s	6,3%
s	5,9%	h	6,1%
i	4,2%	r	6,0%
w	4,0%	d	4,3%
o	3,1%	l	4,0%
f	2,4%	c	2,8%
x	2,1%	u	2,8%
n	2,0%	m	2,4%
d	2,0%	w	2,4%
u	1,9%	f	2,2%
t	1,8%	g	2,0%
k	1,4%	y	2,0%
b	1,3%	p	1,9%
p	1,1%	k	1,8%
y	1,0%	b	1,5%
e	0,9%	v	1,0%
q	0,8%	j	0,3%
a	0,5%	z	0,1%

```
we're no strangers to love
you know the rules and so do i
a full commitment's what i'm thinking of
you wouldn't get this from any other guy
i just wanna tell you how i'm feeling
gotta make you understand
never gonna give you up
never gonna let you down
never gonna run around and desert you
never gonna make you cry
never gonna say goodbye
never gonna tell a lie and hurt you
we've known each other for so long
your heart's been aching, but you're too shy to say it
inside, we both know what's been going on
we know the game and we're gonna play it
and if you ask me how i'm feeling
don't tell me you're too blind to see
never gonna give you up
never gonna let you down
never gonna run around and desert you
never gonna make you cry
never gonna say goodbye
never gonna tell a lie and hurt you
never gonna give you up
never gonna let you down
never gonna run around and desert you
never gonna make you cry
never gonna say goodbye
never gonna tell a lie and hurt you
we've known each other for so long
your heart's been aching, but you're too shy to say it
inside, we both know what's been going on
we know the game and we're gonna play it
i just wanna tell you how i'm feeling
gotta make you understand
```

Frequency analysis

Takeaway:

- The more predictable the cleartext is, the easier it is to crack.
- Frequency analysis requires a lot of text to make sense.
- This stuff was cracked 1200 years ago. Don't use it today.


Zimmermann’s Gamble

Zimmermann's gamble

Arthur Zimmermann, German foreign secretary in 1916.

- For most of the first world war, the US tried to stay out of the conflict
- Following the sinking of the Lusitania in 1915, Germany pledged to only use their submarines on the surface.

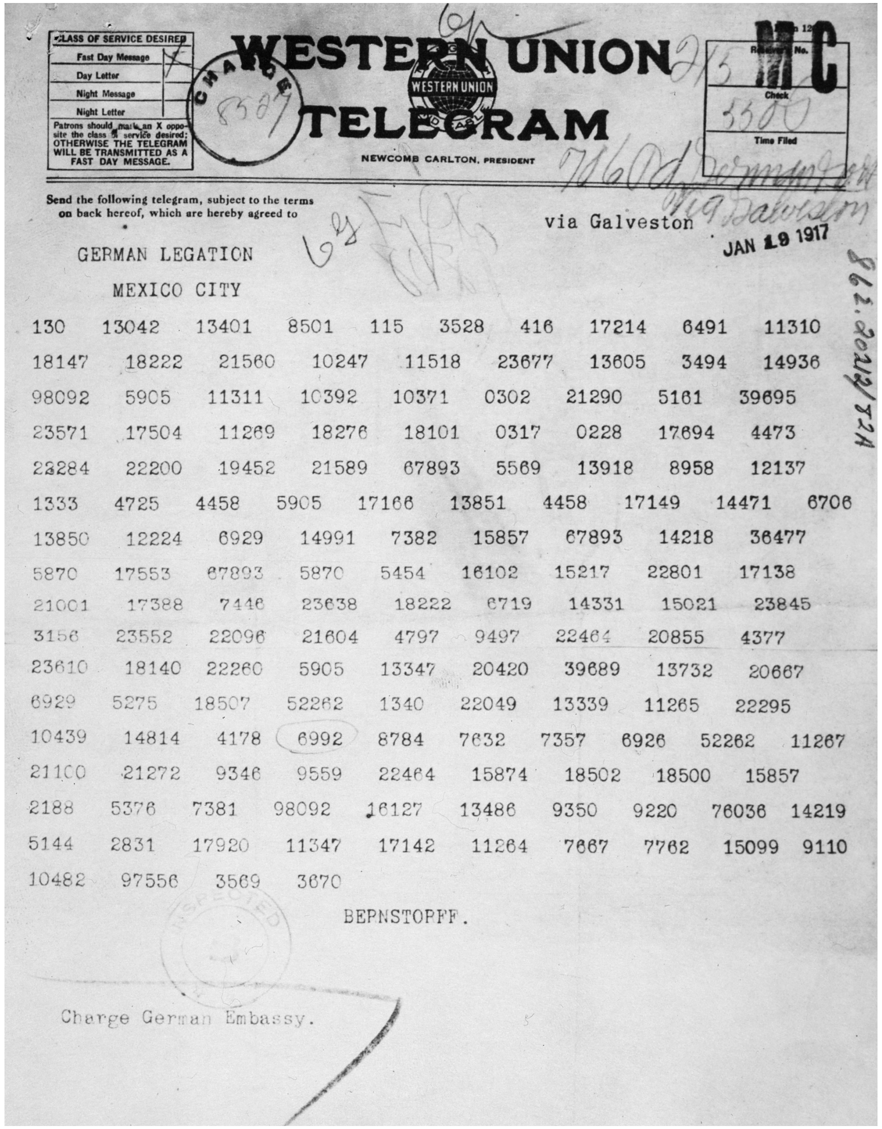
Zimmermann’s gamble

Arthur Zimmermann, German foreign secretary in 1916.

- In 1917, though, Germany thought an unrestricted submarine warfare could be decisive in winning the war.

But they really did not want to pick a fight with the US.
So secretary Zimmermann hatched a plan.

Zimmermann’s gamble



CLASS OF SERVICE DESIRED
Fast Day Message
Day Letter
Night Message
Night Letter
Patrons should mark an X opposite the class of service desired; OTHERWISE THE TELEGRAM WILL BE TRANSMITTED AS A FAST DAY MESSAGE.

WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Receiver's No.
Check
Time Filed

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston

JAN 19 1917

GERMAN LEGATION

MEXICO CITY

130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

BERNSTORFF.

Charge German Embassy.

Zimmermann's gamble

CA.CELED
...tor 1-8-58
...rton, State Dept.
By [illegible]
Date Oct. 22, 1957

TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, ~~[illegible]~~ invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

Zimmermann’s gamble

Arthur Zimmermann, German foreign secretary in 1916.

- Zimmermann publicly admitted in March 1917 that it was genuine.
- The publishing swayed US opinion to declare war only weeks later.

The Enigma

- A polyalphabetic substitution cipher
  - Three scramblers in series.
  - An additional reflector.
  - A plugboard that allows the user to additionally swap up to 6 characters.
- Frequency analysis is effectively useless.
- Reversible!

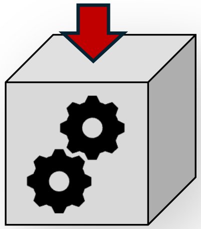
- The breaking of the Enigma was not made public until 1974 – more than 30 years after the fact.


The advent of computers

Modern block ciphers

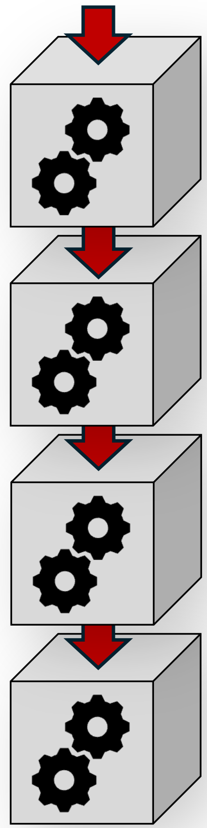
- With the adoption of computer communication, more complex block ciphers started emerging.
- In 1973 the DES algorithm, developed by IBM, was adopted as the national standard in the US.
- DES was, however, made deliberately weak with its 56-bit key.
- During a transitional period, triple-DES was considered adequate, although not great.
- AES has since become a globally accepted standard.

Anatomy of a modern block cipher



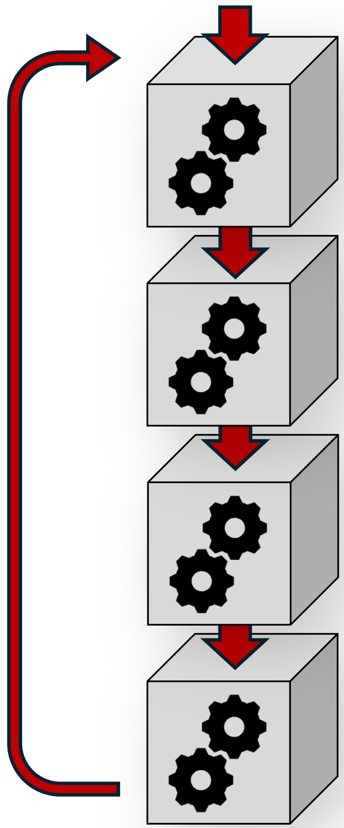
A substitution box (an “S-box”) processes the incoming data frame in a specific way.

Anatomy of a modern block cipher



A substitution box (an “S-box”) processes the incoming data frame in a specific way.

Anatomy of a modern block cipher

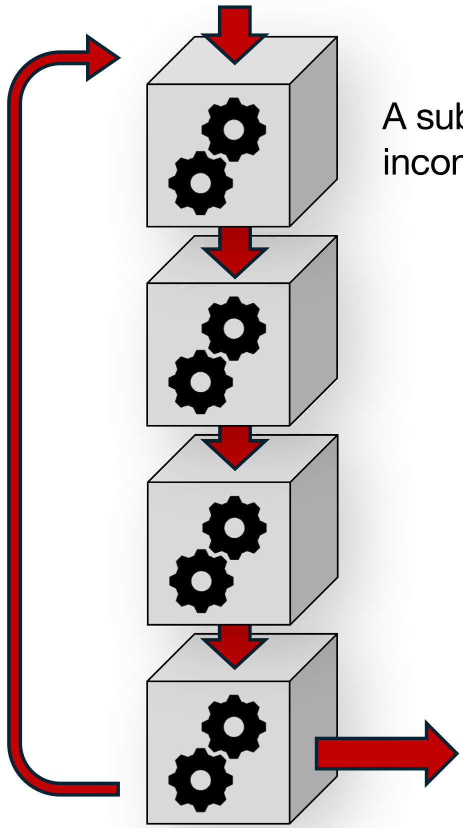


A substitution box (an “S-box”) processes the incoming data frame in a specific way.

This process is repeated for a number of “rounds”.

Anatomy of a modern block cipher

This process is repeated for a number of “rounds”.



A substitution box (an “S-box”) processes the incoming data frame in a specific way.

Ciphertext

AES

Plaintext:	A	t	t	a	c	k		a	t		d	a	w	n	!	
Hex:	41	74	74	61	63	6B	20	61	74	20	64	61	77	6E	21	00

Password:	i	k	n	o	w	w	h	a	t	y	o	u	d	i	d	!
Hex:	69	6B	6E	6F	77	77	68	61	74	79	6F	75	64	69	64	21

AES

Data frame:

41	**74**	**74**	**61**
63	6b	20	61
74	20	64	61
77	6e	21	00

Plaintext:	A	t	t	a	c	k		a	t		d	a	w	n	!	
Hex:	41	74	74	61	63	6B	20	61	74	20	64	61	77	6E	21	00

Password:	i	k	n	o	w	w	h	a	t	y	o	u	d	i	d	!
Hex:	69	6B	6E	6F	77	77	68	61	74	79	6F	75	64	69	64	21

AES

Data frame:

41	**74**	**74**	**61**
63	6b	20	61
74	20	64	61
77	6e	21	00

Key 0:

69	**77**	**74**	**64**
6b	77	79	69
6e	68	6f	64
6f	61	75	21

Key 1:

91	**E6**	**92**	**F6**
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	**f1**	**63**	**95**
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	**ac**
c5	3e
30	8c
44	67

Key expansion

Turns a key into 10, 12 or 14 “round keys”, 16 bytes each.

Password:	**i**	**k**	**n**	**o**	**w**	**w**	**h**	**a**	**t**	**y**	**o**	**u**	**d**	**i**	**d**	**!**
Hex:	69	6B	6E	6F	77	77	68	61	74	79	6F	75	64	69	64	21

AES

Data frame:

Data frame:

41	74	74	61
63	6b	20	61
74	20	64	61
77	6e	21	00

Key 0:

69	77	74	64
6b	77	79	69
6e	68	6f	64
6f	61	75	21

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	ac
c5	3e
30	8c
44	67

Key expansion Turns a key into 10, 12 or 14 “round keys”, 16 bytes each.

AddRoundKey XORs the data frame with the round key

SubBytes

ShiftRows

MixColumns

AES

Data frame:

Data frame:

41	74	74	61
63	6b	20	61
74	20	64	61
[illegible]	[illegible]	21	00

Key 0:

69	77	74	64
6b	77	79	69
6e	68	6f	64
6f	61	75	21

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	a
c5	3
30	8
44	6

XOR: Either one or the other, but not both.

0x41								
0x69								
= **0x28**								

Key expansion — Turns a …, 12 or 14 “round keys”, 16 bytes each.

AddRoundKey — XORs the data frame with the round key

SubBytes

ShiftRows

MixColumns

AES

Data frame:

Data frame:

41	74	74	61
63	6b	20	61
74	20	64	61
		21	00

Key 0:

69	77	74	64
6b	77	79	69
6e	68	6f	64
6f	61	75	21

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

XOR: Either one or the other, but not both.

0x41	0	1	0	0	0	0	0	1
0x69	0	1	1	0	1	0	0	1
= **0x28**								

Key expansion — Turns a … 12 or 14 "round keys", 16 bytes each.

AddRoundKey — XORs the data frame with the round key

SubBytes

ShiftRows

MixColumns

AES

Data frame:

Data frame:

41	74	74	61
63	6b	20	61
74	20	64	61
[illegible]	[illegible]	21	00

Key 0:

69	77	74	64
6b	77	79	69
6e	68	6f	64
6f	61	75	21

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	[illegible]
c5	[illegible]
30	[illegible]
44	[illegible]

XOR: Either one or the other, but not both.

0x41	0	1	0	0	0	0	0	1
0x69	0	1	1	0	1	0	0	1
= **0x28**	0	0	1	0	1	0	0	0

Key expansion — Turns a [illegible], 12 or 14 “round keys”, 16 bytes each.

AddRoundKey — XORs the data frame with the round key

SubBytes

ShiftRows

MixColumns

AES

Data frame:

28	03	00	05
08	1c	59	08
1a	48	0b	05
18	0f	54	21

Data frame:

41	74	74	61
63	6b	20	61
74	20	64	61
77	6e	21	00

Key 0:

69	77	74	64
6b	77	79	69
6e	68	6f	64
6f	61	75	21

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	ac
c5	3e
30	8c
44	67

Key expansion — Turns a key into 10, 12 or 14 “round keys”, 16 bytes each.

AddRoundKey — XORs the data frame with the round key

SubBytes

ShiftRows

MixColumns

AES

Data frame:

34	**7b**	**63**	**6b**
30	9c	cb	30
a2	52	2b	6b
ad	76	20	fd

63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Key 1:

91	**E6**	**92**	**F6**
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	**f1**	**63**	**95**
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	ac
c5	3e
30	8c
44	67

Key expansion

Turns a key into 10, 12 or 14 “round keys”, 16 bytes each.

AddRoundKey

XORs the data frame with the round key

SubBytes

Applies a substitution (S-box) using a static table

ShiftRows

MixColumns

AES

Data frame:

34	7b	63	6b
9c	cb	30	30
2b	6b	a2	52
fd	ad	76	20

Data frame:

			34	7b	63	6b
		30	9c	cb	30	
	a2	52	2b	6b		
ad	76	20	fd			

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	ac
c5	3e
30	8c
44	67

Key expansion Turns a key into 10, 12 or 14 “round keys”, 16 bytes each.

AddRoundKey XORs the data frame with the round key

SubBytes Applies a substitution (S-box) using a static table

ShiftRows Shifts the rows in the data frame

MixColumns

AES

Data frame:

34	**7b**	**63**	**6b**
9c	cb	30	30
2b	6b	a2	52
fd	ad	76	20

Data frame:

34	**7b**	**63**	**6b**
9c	cb	30	30
2b	6b	a2	52
fd	ad	76	20

$$\times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Key 1:

91	**E6**	**92**	**F6**
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	**f1**	**63**	**95**
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3

5c	**ac**
c5	3e
30	8c
44	67

Key expansion — Turns a key into 10, 12 or 14 "round keys", 16 bytes each.

AddRoundKey — XORs the data frame with the round key

SubBytes — Applies a substitution (S-box) using a static table

ShiftRows — Shifts the rows in the data frame

MixColumns — Uses matrix multiplication to create a substitution

AES

Data frame:

Data frame:

41	74	74	61
63	6b	20	61
74	20	64	61
77	6e	21	00

Key 1:

91	E6	92	F6
28	5f	26	4f
93	Fb	94	F0
2c	4d	38	19

Key 2:

17	f1	63	95
a4	fb	dd	92
47	bc	28	d8
6e	23	1b	02

Key 3:

5c	ad	ce	5b
c5	3e	e3	71
30	8c	a4	7c
44	67	7c	7e

Key 4

Key expansion — Turns a key into 10, 12 or 14 "round keys", 16 bytes each.

Repeat 9, 11, or 13 times

AddRoundKey — XORs the data frame with the round key

SubBytes — Applies a substitution (S-box) using a static table

ShiftRows — Shifts the rows in the data frame

MixColumns — Uses matrix multiplication to create a substitution

AES

Used for:

- Data encryption for TLS (formerly SSL)
- Full disk encryption

- AES is ubiquitous - found in
  - software,
  - operating systems,
  - CPUs (including with native instructions),
  - PLCs, storage controllers, etc
  - USB flash drives

- The design allows for impressively high throughput

Hashing

Encryption

- Encryption generates "ciphertext".

```
Please pay the recipient $100.

09urxmo298r2eofwehfi2ehfoi2efh
```

Hashing

- Hashing creates a "digest".

```
Please pay the recipient $100.

0x4273a48237e9280c
```

Hashing

Encryption	Hashing
• Encryption generates "ciphertext".	• Hashing creates a "digest".
• Encryption is *reversible*. One ciphertext generates exactly one plaintext.	• Hashing is *not reversible*. Sometimes, the two plaintexts can generate the same digest

```
Please pay the recipient $100.

09urxmo298r2eofwehfi2ehfoi2efh
```

```
Please pay the recipient $100.

0x4273a48237e9280c
```

Hashing

Encryption

- Encryption generates "ciphertext".
- Encryption is *reversible*. One ciphertext generates exactly one plaintext.
- Encryption uses a key or password

```
Please pay the recipient $100.

09urxmo298r2eofwehfi2ehfoi2efh
```

Hashing

- Hashing creates a "digest".
- Hashing is *not reversible*. Sometimes, the two plaintexts can generate the same digest
- Hashing can use a "salt"

```
Please pay the recipient $100.

0x4273a48237e9280c
```

Hashing

Encryption	Hashing
Encryption generates “ciphertext”.	Hashing creates a “digest”.
Encryption is *reversible*. One ciphertext generates exactly one plaintext.	Hashing is *not reversible*. Sometimes, the two plaintexts can generate the same digest
Encryption uses a key or password	Hashing can use a “salt”
Length of ciphertext is similar in length to plaintext	Digests are typically fixed-width

```
Please pay the recipient $100.

09urxmo298r2eofwehfi2ehfoi2efh
```

```
Please pay the recipient $100.

0x4273a48237e9280c
```

Hashing

Hashing

passw0rd#1!

0x138d20ec87452a0f

Hashing

Hashing

passw0rd#1!

0x138d20ec87452a0f

0x138d20ec87452a0f

passw0rd#1!

There are "**rainbow tables**" for common passwords, allowing an attacker to reverse-engineer hashes.

Hashing

Hashing	Hashing with salt
passw0rd#1!	Salt
0x138d20ec87452a0f	xyzpassw0rd#1!
0x138d20ec87452a0f	0x39ac862c958d3460
passw0rd#1!	

Hashing

Hashing	Hashing with salt

passw0rd#1!

0x138d20ec87452a0f

0x138d20ec87452a0f

passw0rd#1!

Salt

xyzpassw0rd#1!

0x39ac862c958d3460

Different salt

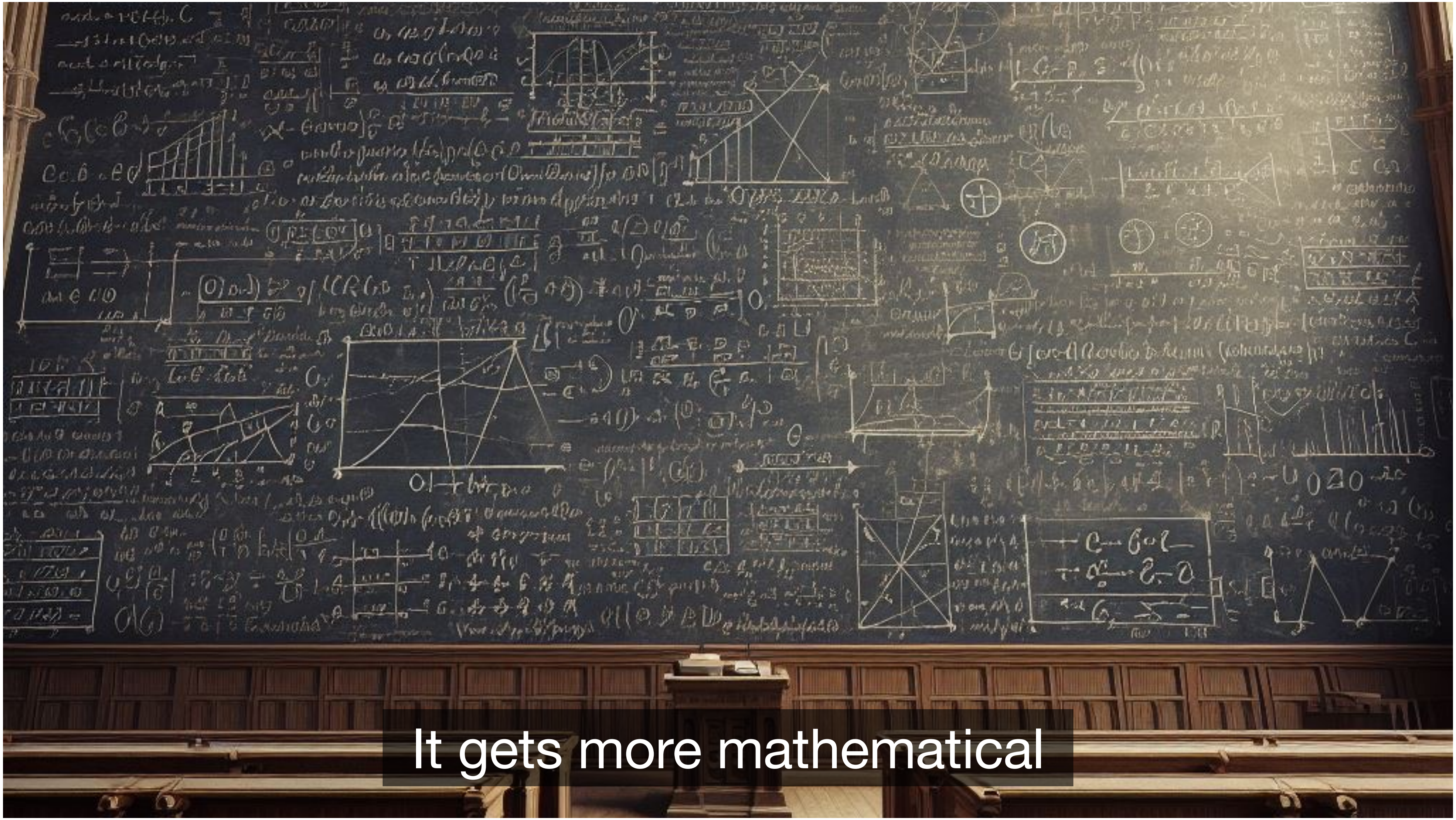
abcpassw0rd#1!

0xaa1209752f83942b

Hashing

How are hashes used?

- git commits
- Storing passwords
- Detecting changes in code/text, etc
- Digital signatures


It gets more mathematical

Asymmetric encryption

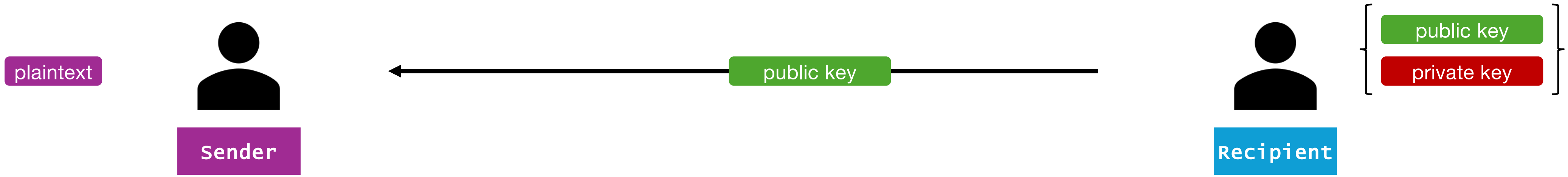
Also called "public key encryption".

The two most common algorithms are

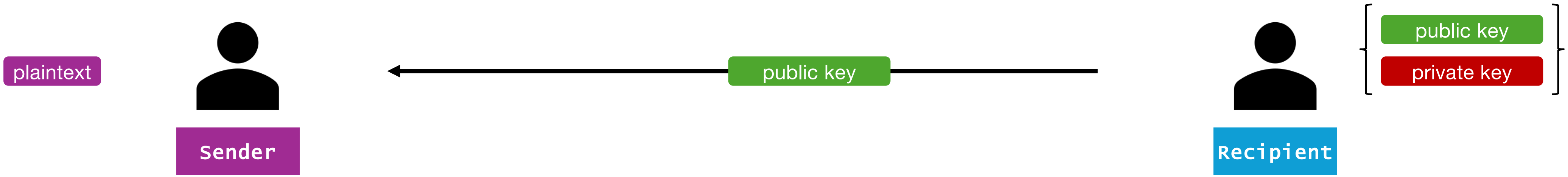
- RSA (Rivest, Shamir, Adleman)
- Elliptic Curve Encryption (ECC)

Asymmetric encryption

Asymmetric encryption

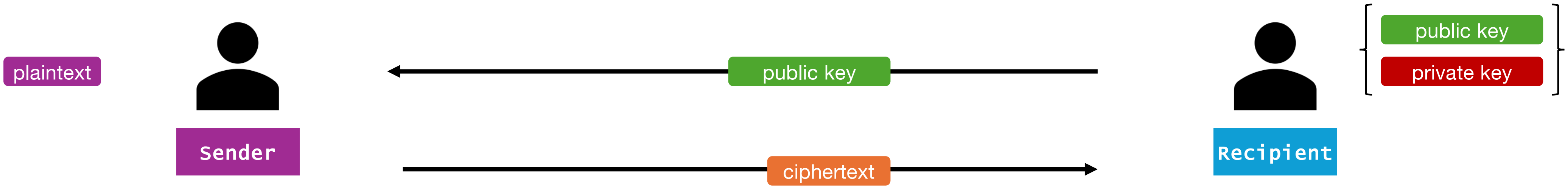


Asymmetric encryption



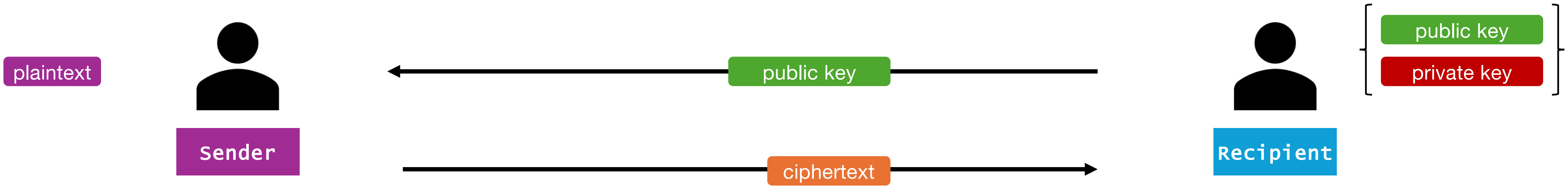
ciphertext $= fn\,($ plaintext $,$ public key $)$

Asymmetric encryption



ciphertext $= fn\,($ plaintext , public key $)$

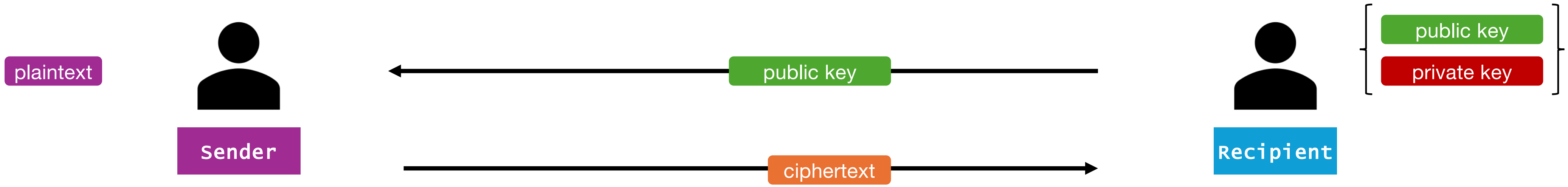
Asymmetric encryption



ciphertext $= fn\,($ plaintext , public key $)$

plaintext $= fn\,($ ciphertext , private key $)$

Asymmetric encryption



ciphertext $= fn$ (plaintext , public key)

plaintext $= fn$ (ciphertext , private key)

This ”one-way” function is fairly simple math. But the reverse is very difficult.

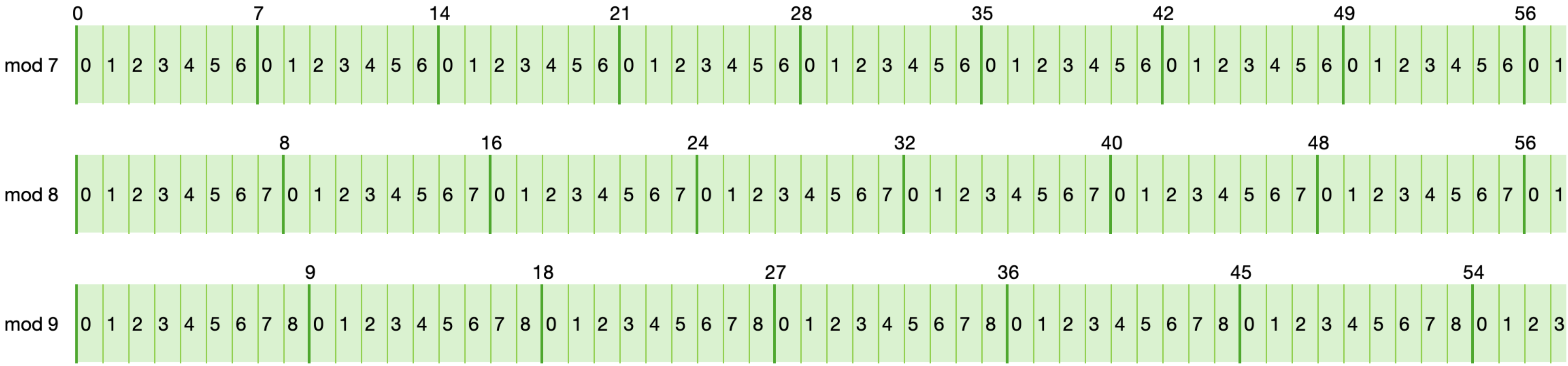
Jevons' number

"Can the reader say what two numbers multiplied together will produce the number 8,616,460,799? I think it unlikely that anyone but myself will ever know."

William Stanley Jevons
Principles of Science, 1874

Some required math basics

The **modulo** of an integer is the ”remainder” when you divide the integer by the modulo.



Think of the modulo as an incrementing number that resets at intervals.

Some required math basics

A **prime** number is an integer that is evenly divisible by only itself and 1.

1	1 is **prime**	9	9 = 3·3	17	17 is **prime**
2	2 is **prime**	10	10 = 2·5	18	18 = 2·3·3
3	3 is **prime**	11	11 is **prime**	19	19 is **prime**
4	4 = 2·2	12	12 = 2·2·3	20	20 = 2·2·5
5	5 is **prime**	13	13 is **prime**	21	21 = 3·7
6	6 = 2·3	14	14 = 7·2	22	22 = 11·2
7	7 is **prime**	15	15 = 5·3	23	23 is **prime**
8	8 = 2·2·2	16	16 = 2·2·2·2	24	24 = 2·2·2·3

Every non-prime integer is a multiple of two or more prime numbers.

RSA

p = 247 193 PRIVATE
q = 395 767 PRIVATE

p·q = 97 830 832 031 (equivalent of a 37-bit key size) PUBLIC

RSA

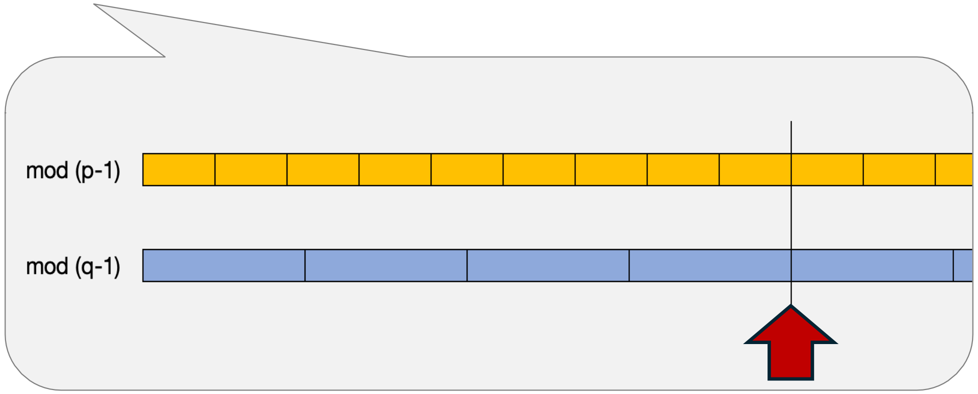
$p = 247\ 193$ PRIVATE

$q = 395\ 767$ PRIVATE

$p \cdot q = 97\ 830\ 832\ 031$ PUBLIC

The least common multiple of the two prime factors.

$\lambda(p \cdot q) = \mathrm{lcm}(p-1, q-1) = 48\ 915\ 094\ 536$



RSA

$p = 247\ 193$ PRIVATE
$q = 395\ 767$ PRIVATE

$p \cdot q = 97\ 830\ 832\ 031$ PUBLIC

$\lambda(p \cdot q) = \mathrm{lcm}(p-1,\ q-1) = 48\ 915\ 094\ 536$

e is chosen - must be coprime with $\lambda(p \cdot q)$

$e = 65\ 537$ PUBLIC

Two numbers are said to be coprime when they don't share any divisors greater than 1.

3 and 8 are coprime.

4 and 6 are not coprime, because both can be evenly divided by 2.

RSA

$p = 247\ 193$ PRIVATE

$q = 395\ 767$ PRIVATE

$p \cdot q = 97\ 830\ 832\ 031$ PUBLIC

$\lambda(p \cdot q) = \mathrm{lcm}(p-1,\ q-1) = 48\ 915\ 094\ 536$

$e = 65\ 537$ PUBLIC

$d = 1\ 173\ 299\ 489$ PRIVATE

d is a function of
e and λ(p·q)

RSA

Public key

e p·q

Private key

d p·q

p = 247 193 PRIVATE

q = 395 767 PRIVATE

p·q = 97 830 832 031 PUBLIC

λ(p·q) = lcm(p-1, q-1) = 48 915 094 536

e = 65 537 PUBLIC

d = 1 173 299 489 PRIVATE

RSA

Public key

e | p·q

Private key

d | p·q

$p = 247\ 193$ PRIVATE

$q = 395\ 767$ PRIVATE

$p \cdot q = 97\ 830\ 832\ 031$ PUBLIC

$\lambda(p \cdot q) = \mathrm{lcm}(p-1, q-1) = 48\ 915\ 094\ 536$

$e = 65\ 537$ PUBLIC

$d = 1\ 173\ 299\ 489$ PRIVATE

plaintext $m = 1\ 234\ 567$

RSA

Public key

e p·q

Private key

d p·q

$p = 247\,193$ PRIVATE

$q = 395\,767$ PRIVATE

$p \cdot q = 97\,830\,832\,031$ PUBLIC

$\lambda(p \cdot q) = \text{lcm}(p-1, q-1) = 48\,915\,094\,536$

$e = 65\,537$ PUBLIC

$d = 1\,173\,299\,489$ PRIVATE

plaintext $m = 1\,234\,567$

Encryption: ciphertext $c = m^{e} \bmod p \cdot q = 65\,699\,382\,295$

RSA

Public key

e p·q

Private key

d p·q

$p = 247\,193$ PRIVATE

$q = 395\,767$ PRIVATE

$p \cdot q = 97\,830\,832\,031$ PUBLIC

$\lambda(p \cdot q) = \mathrm{lcm}(p-1, q-1) = 48\,915\,094\,536$

$e = 65\,537$ PUBLIC

$d = 1\,173\,299\,489$ PRIVATE

plaintext $m = 1\,234\,567$

Encryption: ciphertext $c = m^{e} \bmod p \cdot q = 65\,699\,382\,295$

Decryption: plaintext $m = c^{d} \bmod p \cdot q = 1\,234\,567$

RSA

Public key

e p·q

Private key

d p·q

$p = 247\ 193$ PRIVATE

$q = 395\ 767$ PRIVATE

EASY

$p \cdot q = 97\ 830\ 832\ 031$ PUBLIC

$\lambda(p \cdot q) = \mathrm{lcm}(p-1, q-1) = 48\ 915\ 094\ 536$

EASY

$e = 65\ 537$ PUBLIC

EASY

$d = 1\ 173\ 299\ 489$ PRIVATE

plaintext $m = 1\ 234\ 567$

Encryption: ciphertext $c = m^{\boxed{e}} \bmod \boxed{p \cdot q} = 65\ 699\ 382\ 295$

Decryption: plaintext $m = c^{\boxed{d}} \bmod \boxed{p \cdot q} = 1\ 234\ 567$

RSA

Public key

e p·q

Private key

d p·q

$p = 247\ 193$ PRIVATE

$q = 395\ 767$ PRIVATE

VERY DIFFICULT

EASY

$p \cdot q = 97\ 830\ 832\ 031$ PUBLIC

$\lambda(p \cdot q) = \mathrm{lcm}(p-1, q-1) = 48\ 915\ 094\ 536$

EASY

$e = 65\ 537$ PUBLIC

EASY

$d = 1\ 173\ 299\ 489$ PRIVATE

plaintext $m = 1\ 234\ 567$

Encryption: ciphertext $c = m^{\boxed{e}} \bmod \boxed{p \cdot q} = 65\ 699\ 382\ 295$

Decryption: plaintext $m = c^{\boxed{d}} \bmod \boxed{p \cdot q} = 1\ 234\ 567$

RSA

Public key

e p·q

Private key

d p·q

EASY

EASY

EASY

$p = 247\,193$ PRIVATE

$q = 395\,767$ PRIVATE

VERY DIFFICULT

$p \cdot q = 97\,830\,832\,031$ PUBLIC

VERY DIFFICULT

$\lambda(p \cdot q) = \mathrm{lcm}(p-1, q-1) = 48\,915\,094\,536$

$e = 65\,537$ PUBLIC

$d = 1\,173\,299\,489$ PRIVATE

plaintext $m = 1\,234\,567$

Encryption: ciphertext $c = m^{\boxed{e}} \bmod \boxed{p \cdot q} = 65\,699\,382\,295$

Decryption: plaintext $m = c^{\boxed{d}} \bmod \boxed{p \cdot q} = 1\,234\,567$

RSA

Which two prime numbers multiply to form this result?

p·q = 307 557 980 540 279 294 232 889 703 319 188 128 660 178 473 934 208 428 941 529 507 290 014 700 153 668 237 193 840 016 959 648 364
895 607 412 313 512 951 227 128 221 045 224 559 205 896 270 620 516 337 082 808 739 837 588 961 254 353 255 830 027 840 166 730 656 182
059 939 220 104 108 619 578 923 418 753 353 272 689 302 172 594 198 324 164 702 665 735 456 123 372 112 931 626 325 928 229 001 407 813
673 728 165 736 146 453 855 297 160 675 344 337 682 232 580 861 895 550 087 017 268 994 973 859 666 439 292 146 128 040 198 542 069 704
950 256 297 882 300 231 190 908 477 379 997 770 136 082 327 312 952 935 689 262 892 629 588 356 961 572 827 315 993 037 083 976 288 531
844 687 385 055 499 903 044 511 450 039 012 753 955 234 146 300 919 159 567 440 161 336 043 261 154 130 838 051 072 922 104 944 728 496
403 982 179 433 674 737 196 227 320 743 852 387 755 112 827 166 448 480 967 725 082 114 732 889 029 922 825 155 341 072 387 200 684 641
384 860 543 585 552 397 175 025 316 086 544 491 738 636 866 945 634 642 543 579 069 450 381 289 326 765 766 927 135 738 548 449 509 733
934 196 182 156 369 067 294 637 972 434 294 530 583 620 547 004 745 403 650 343 545 789 619 780 447 563 019 123 590 591

(roughly a 4096 bit key size)

RSA

Which two prime numbe

p·q = 307 557 980 540 279 294 … 0 016 959 648 364
895 607 412 313 512 951 227 12 … 166 730 656 182
059 939 220 104 108 619 578 92 … 229 001 407 813
673 728 165 736 146 453 855 29 … 198 542 069 704
950 256 297 882 300 231 190 90 … 083 976 288 531
844 687 385 055 499 903 044 51 … 104 944 728 496
403 982 179 433 674 737 196 22 … 387 200 684 641
384 860 543 585 552 397 175 02 … 548 449 509 733
934 196 182 156 369 067 294 63

(roughly a 4096 bit key s

You

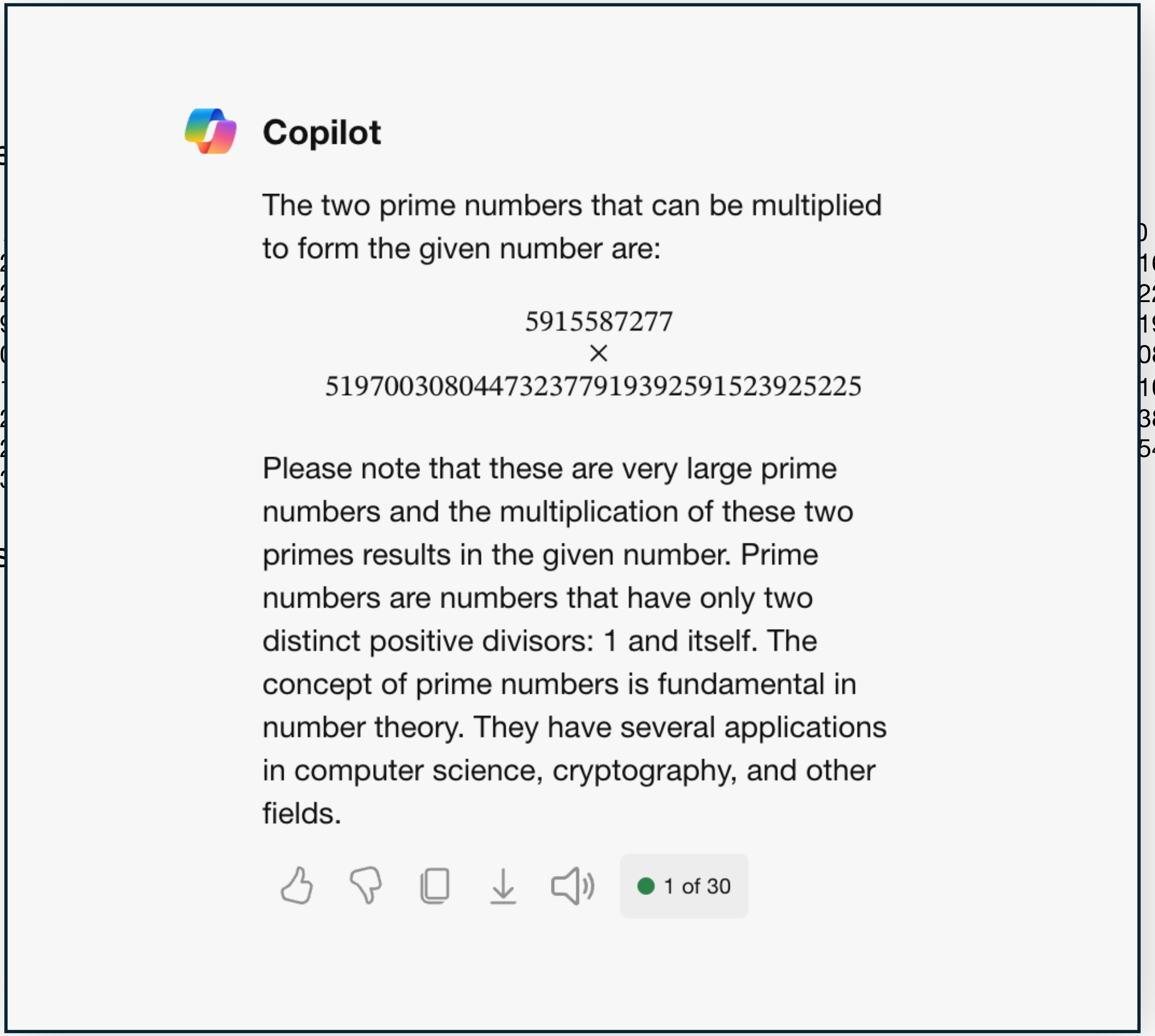
Which two prime numbers can be multiplied to form 307 557 980 540 279 294 232 889 703 319 188 128 660 178 473 934 208 428 941 529 507 290 014 700 153 668 237 193 840 016 959 648 364 895 607 412 313 512 951 227 128 221 045 224 559 205 896 270 620 516 337 082 808 739 837 588 961 254 353 255 830 027 840 166 730 656 182 059 939 220 104 108 619 578 923 418 753 353 272 689 302 172 594 198 324 164 702 665 735 456 123 372 112 931 626 325 928 229 001 407 813 673 728 165 736 146 453 855 297 160 675 344 337 682 232 580 861 895 550 087 017 268 994 973 859 666 439 292 146 128 040 198 542 069 704 950 256 297 882 300 231 190 908 477 379 997 770 136 082 327 312 952 935 689 262 892 629 588 356 961 572 827 315 993 037 083 976 288 531 844 687 385 055 499 903 044 511 450 039 012 753 955 234 146 300 919 159 567 440 161 336 043 261 154 130 838 051 072 922 104 944 728 496 403 982 179 433 674 737 196 227 320 743 852 387 755 112 827 166 448 480 967 725 082 114 732 889 029 922 825 155 341 072 387 200 684 641 384 860 543 585 552 397 175 025 316 086 544 491 738 636

RSA

Which two prime numbe

p·q = 307 557 980 540 279 294 … 0 016 959 648 364
895 607 412 313 512 951 227 12 … 166 730 656 182
059 939 220 104 108 619 578 92 … 229 001 407 813
673 728 165 736 146 453 855 29 … 198 542 069 704
950 256 297 882 300 231 190 90 … 083 976 288 531
844 687 385 055 499 903 044 51 … 104 944 728 496
403 982 179 433 674 737 196 22 … 387 200 684 641
384 860 543 585 552 397 175 02 … 548 449 509 733
934 196 182 156 369 067 294 63

(roughly a 4096 bit key s



Copilot

The two prime numbers that can be multiplied to form the given number are:

$$5915587277 \times 5197003080447323779193925915239 25225$$

Please note that these are very large prime numbers and the multiplication of these two primes results in the given number. Prime numbers are numbers that have only two distinct positive divisors: 1 and itself. The concept of prime numbers is fundamental in number theory. They have several applications in computer science, cryptography, and other fields.

1 of 30

RSA

Which two prime numbe

p·q = 307 557 980 540 279 294
895 607 412 313 512 951 227 12
059 939 220 104 108 619 578 92
673 728 165 736 146 453 855 29
950 256 297 882 300 231 190 90
844 687 385 055 499 903 044 51
403 982 179 433 674 737 196 22
384 860 543 585 552 397 175 02
934 196 182 156 369 067 294 63

0 016 959 648 364
166 730 656 182
229 001 407 813
198 542 069 704
083 976 288 531
104 944 728 496
387 200 684 641
548 449 509 733

(roughly a 4096 bit key s

RSA

Which two prime numbers multiply to form this result?

p·q = 307 557 980 540 279 294 232 889 703 319 188 128 660 178 473 934 208 428 941 529 507 290 014 700 153 668 237 193 840 016 959 648 364 895 607 412 313 512 951 227 128 221 045 224 559 205 896 270 620 516 337 082 808 739 837 588 961 254 353 255 830 027 840 166 730 656 182 059 939 220 104 108 619 578 923 418 753 353 272 689 302 172 594 198 324 164 702 665 735 456 123 372 112 931 626 325 928 229 001 407 813 673 728 165 736 146 453 855 297 160 675 344 337 682 232 580 861 895 550 087 017 268 994 973 859 666 439 292 146 128 040 198 542 069 704 950 256 297 882 300 231 190 908 477 379 997 770 136 082 327 312 952 935 689 262 892 629 588 356 961 572 827 315 993 037 083 976 288 531 844 687 385 055 499 903 044 511 450 039 012 753 955 234 146 300 919 159 567 440 161 336 043 261 154 130 838 051 072 922 104 944 728 496 403 982 179 433 674 737 196 227 320 743 852 387 755 112 827 166 448 480 967 725 082 114 732 889 029 922 825 155 341 072 387 200 684 641 384 860 543 585 552 397 175 025 316 086 544 491 738 636 866 945 634 642 543 579 069 450 381 289 326 765 766 927 135 738 548 449 509 733 934 196 182 156 369 067 294 637 972 434 294 530 583 620 547 004 745 403 650 343 545 789 619 780 447 563 019 123 590 591

The correct answer is

p = 5 357 543 035 931 336 604 742 125 245 300 009 052 807 024 058 527 668 037 218 751 941 851 755 255 624 680 612 465 991 894 078 479 290 637 973 364 587 765 734 125 935 726 428 461 570 217 992 288 787 349 287 401 967 283 887 412 115 492 710 537 302 531 185 570 938 977 091 076 523 237 491 790 970 633 699 383 779 582 771 973 038 531 457 285 598 238 843 271 083 830 214 915 826 312 193 418 602 834 035 927

and

q = 57 406 534 763 712 726 211 641 660 058 884 099 201 115 885 104 434 760 023 882 136 841 288 313 069 618 515 692 832 974 315 825 313 495 922 298 231 949 373 138 672 355 948 043 152 766 571 296 567 808 332 659 269 564 994 572 656 140 000 344 389 574 120 022 435 714 463 495 031 743 122 390 807 731 823 194 181 973 658 513 020 233 176 985 452 498 279 081 199 404 472 314 802 811 655 824 768 082 110 985 166 340 672 084 454 492 229 252 801 189 742 403 957 029 450 467 388 250 214 501 358 353 312 915 261 004 066 118 140 645 880 633 941 658 603 299 497 698 209 063 510 889 929 202 021 079 926 591 625 770 444 716 951 045 960 277 478 891 794 836 019 580 040 978 608 315 291 377 690 212 791 863 007 764 174 393 209 716 027 254 457 637 891 941 312 587 717 764 400 411 421 385 408 982 726 881 092 425 574 515 033

RSA

Which two prime numbers multiply to form this result?

p·q = 307 557 980 540 279 2… …7 193 840 016 959 648 364
895 607 412 313 512 951 227… …027 840 166 730 656 182
059 939 220 104 108 619 578… …325 928 229 001 407 813
673 728 165 736 146 453 855… …128 040 198 542 069 704
950 256 297 882 300 231 190… …993 037 083 976 288 531
844 687 385 055 499 903 044… …072 922 104 944 728 496
403 982 179 433 674 737 196… …341 072 387 200 684 641
384 860 543 585 552 397 175… …135 738 548 449 509 733
934 196 182 156 369 067 294… …590 591

The correct answer is …

p = 5 357 543 035 931 336 6… …2 465 991 894 078 479 290
637 973 364 587 765 734 125… …531 185 570 938 977 091
076 523 237 491 790 970 633… …418 602 834 035 927

and

q = 57 406 534 763 712 726 2… …92 832 974 315 825 313 495
922 298 231 949 373 138 672… …120 022 435 714 463 495
031 743 122 390 807 731 823 194 181 973 658 513 020 233 176 985 452 498 279 081 199 404 472 314 802 811 655 824 768 082 110 985 166 340
672 084 454 492 229 252 801 189 742 403 957 029 450 467 388 250 214 501 358 353 312 915 261 004 066 118 140 645 880 633 941 658 603 299
497 698 209 063 510 889 929 202 021 079 926 591 625 770 444 716 951 045 960 277 478 891 794 836 019 580 040 978 608 315 291 377 690 212
791 863 007 764 174 393 209 716 027 254 457 637 891 941 312 587 717 764 400 411 421 385 408 982 726 881 092 425 574 515 033

If you guessed correctly,

just shake your head and smile.

And never, ever tell a soul.

RSA

RSA has a published list of prime factors, some of which have prize money attached.

The largest RSA number factored so far is ”RSA-250”, equivalent of a 829 bit key, which was factored in 2020.

The computation effort is roughly equivalent to 2700 CPU core years on a 2.1 GHz Xeon.

The patent number for RSA, 4 405 829, is actually prime.

Elliptic Curve (ECC)

Way, *waaaaay* too complicated for an introductory session.

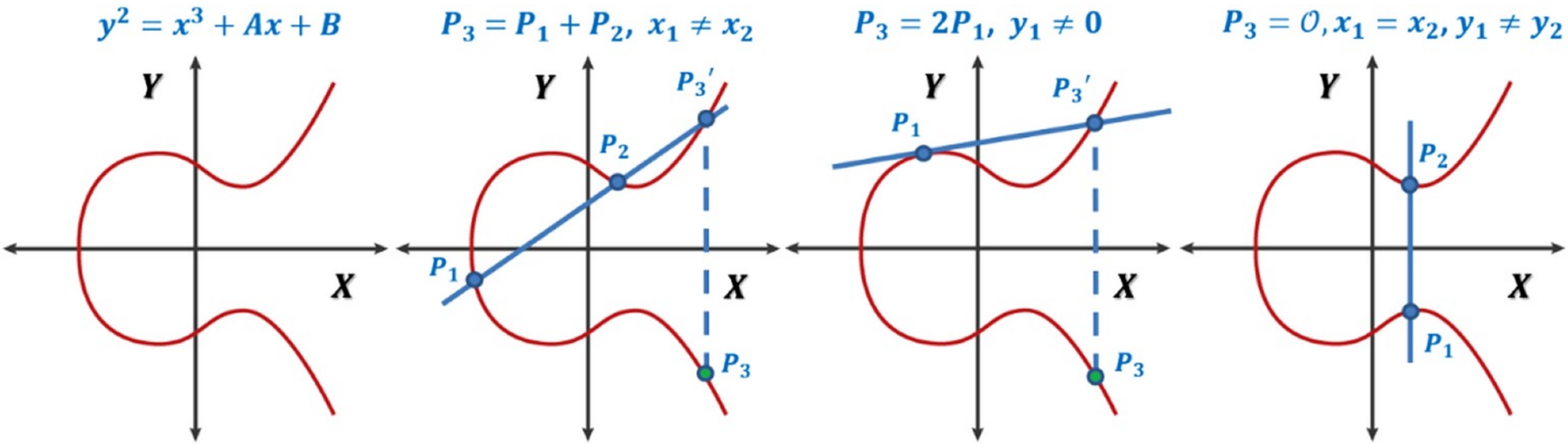


Image source: https://www.nature.com/articles/s41598-022-17045-x

RSA & ECC: Asymmetric encryption

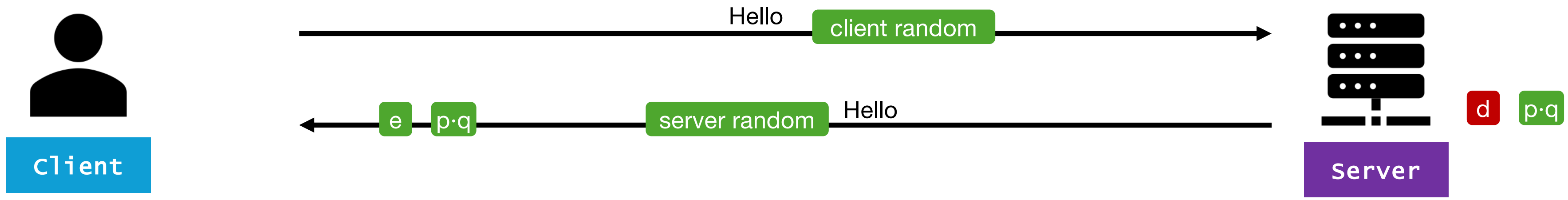
How is asymmetric encryption used?

- Key exchange for TLS (formerly SSL), SSH
- Blockchains, crypto currencies
- Secure signatures

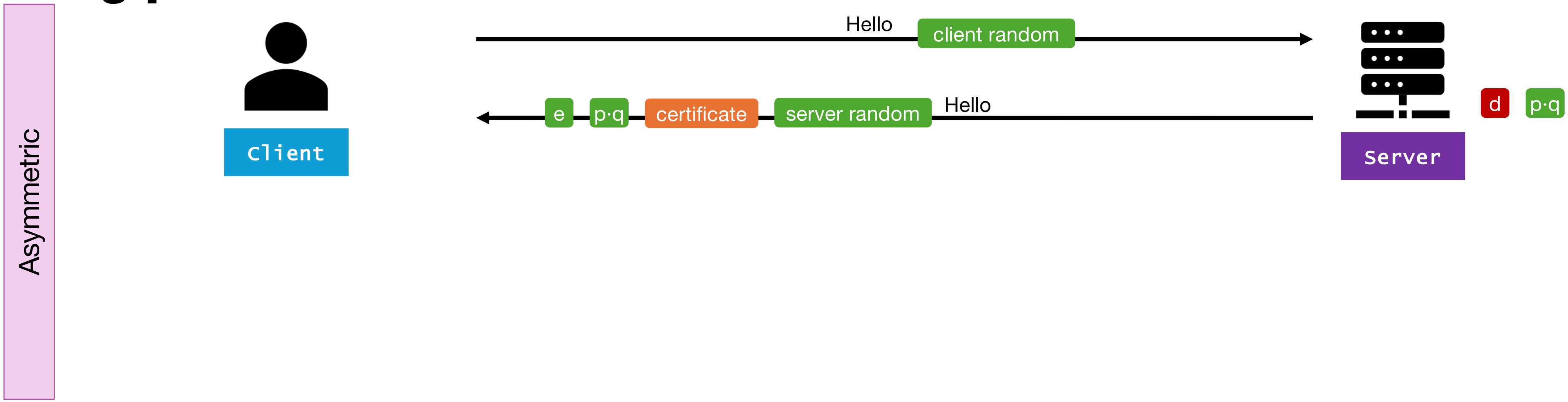
Asymmetric algorithms are typically only used to encrypt the key or hash digest. The data is encrypted using much faster symmetric encryption.

Big picture

Asymmetric

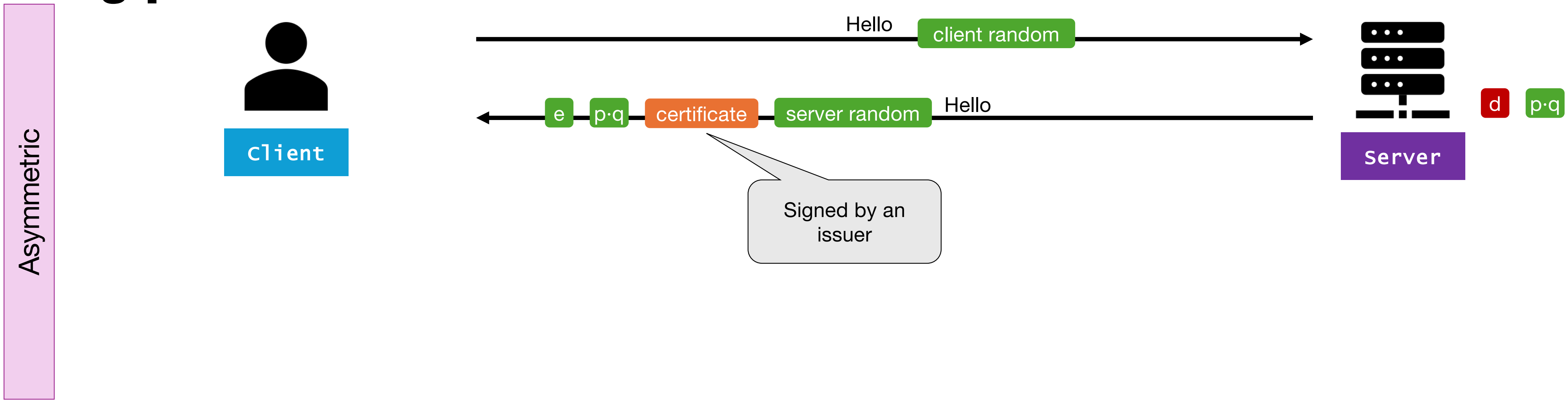


Big picture



Asymmetric
Client
Hello
client random
e
p·q
certificate
server random
Hello
d
p·q
Server

Big picture


Asymmetric
Client
Hello
client random
e
p·q
certificate
server random
Hello
Signed by an issuer
Server
d
p·q

Signatures

Signing uses the same function as encryption and decryption, but *e* and *d* are swapped.

ciphertext $c = m^{e} \bmod p \cdot q$

plaintext $m = c^{d} \bmod p \cdot q$

Signatures

ciphertext $c = m^{e} \bmod p \cdot q$

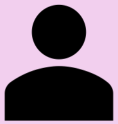
plaintext $m = c^{d} \bmod p \cdot q$

Signer

digest = hash(m)

Signatures

Signature proves that you have private key d.



Signer

digest = hash(m)

signature $s = \text{digest}^{d} \bmod p \cdot q$

ciphertext $c = m^{e} \bmod p \cdot q$

plaintext $m = c^{d} \bmod p \cdot q$

Signatures

ciphertext $c = m^{e} \bmod p \cdot q$

plaintext $m = c^{d} \bmod p \cdot q$

Signature proves that you have private key d.

Signer

digest = hash(m)

signature $s = \text{digest}^{d} \bmod p \cdot q$

Reader

digest = hash(m)

Signatures

ciphertext $c = m^{e} \bmod p \cdot q$

plaintext $m = c^{d} \bmod p \cdot q$

Signature proves that you have private key d.

Signer

$\text{digest} = \text{hash}(m)$

$\text{signature } s = \text{digest}^{d} \bmod p \cdot q$

Only needs the public key e to verify the signature.

Reader

$\text{digest} = \text{hash}(m)$

$\text{verify that digest} = \text{signature } s^{e} \bmod p \cdot q$

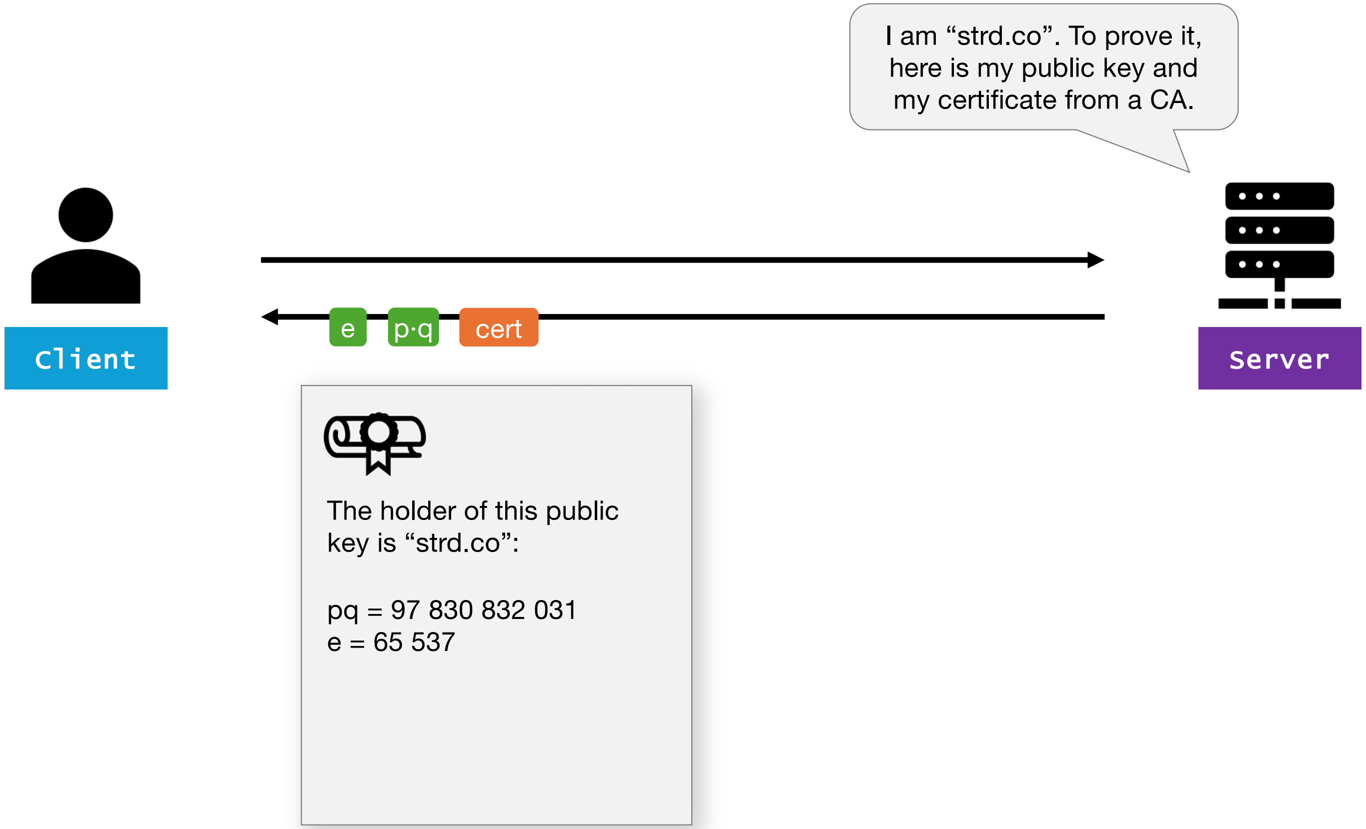
Signatures

How are cryptographic signatures used?

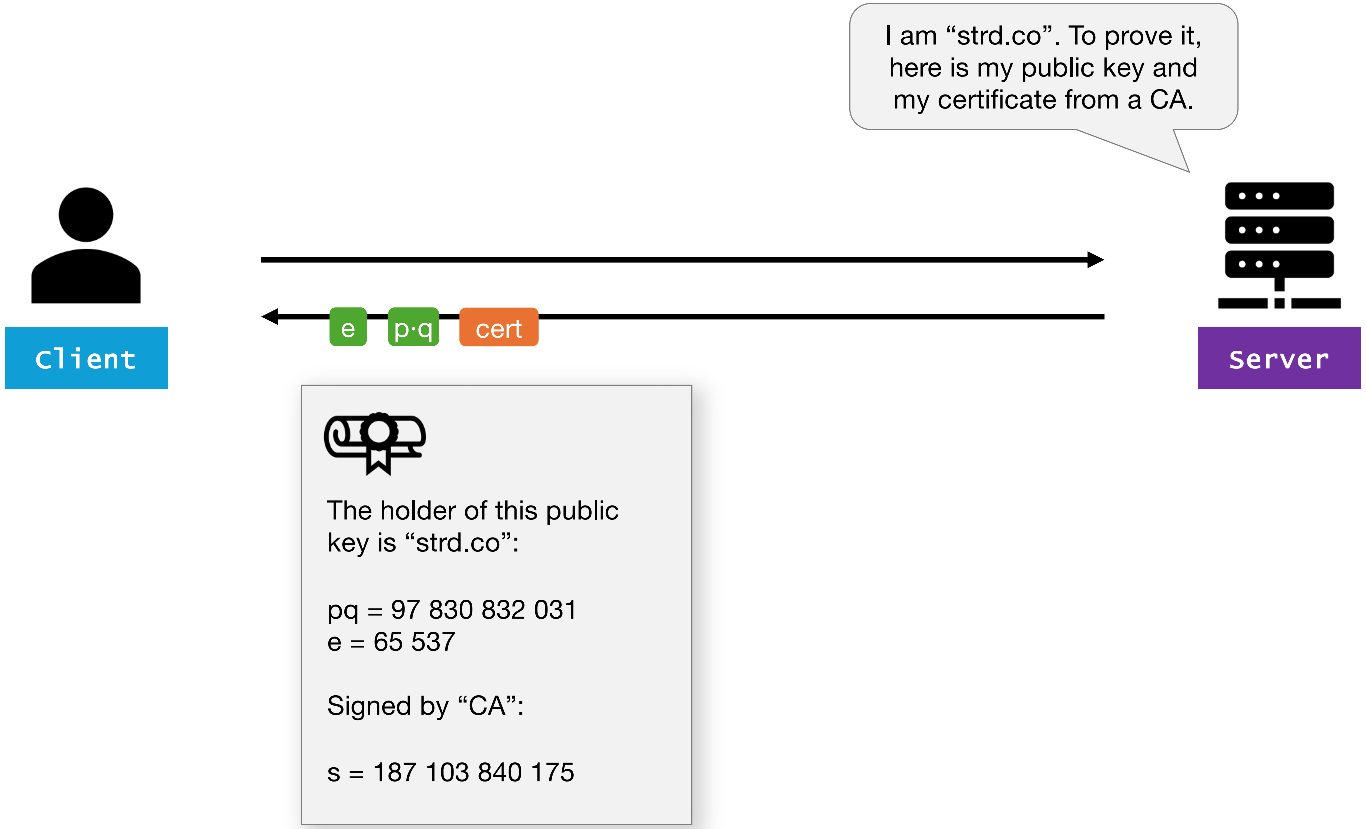
- Code signing
- DNSSEC
- Signing certificates

Certificate chains

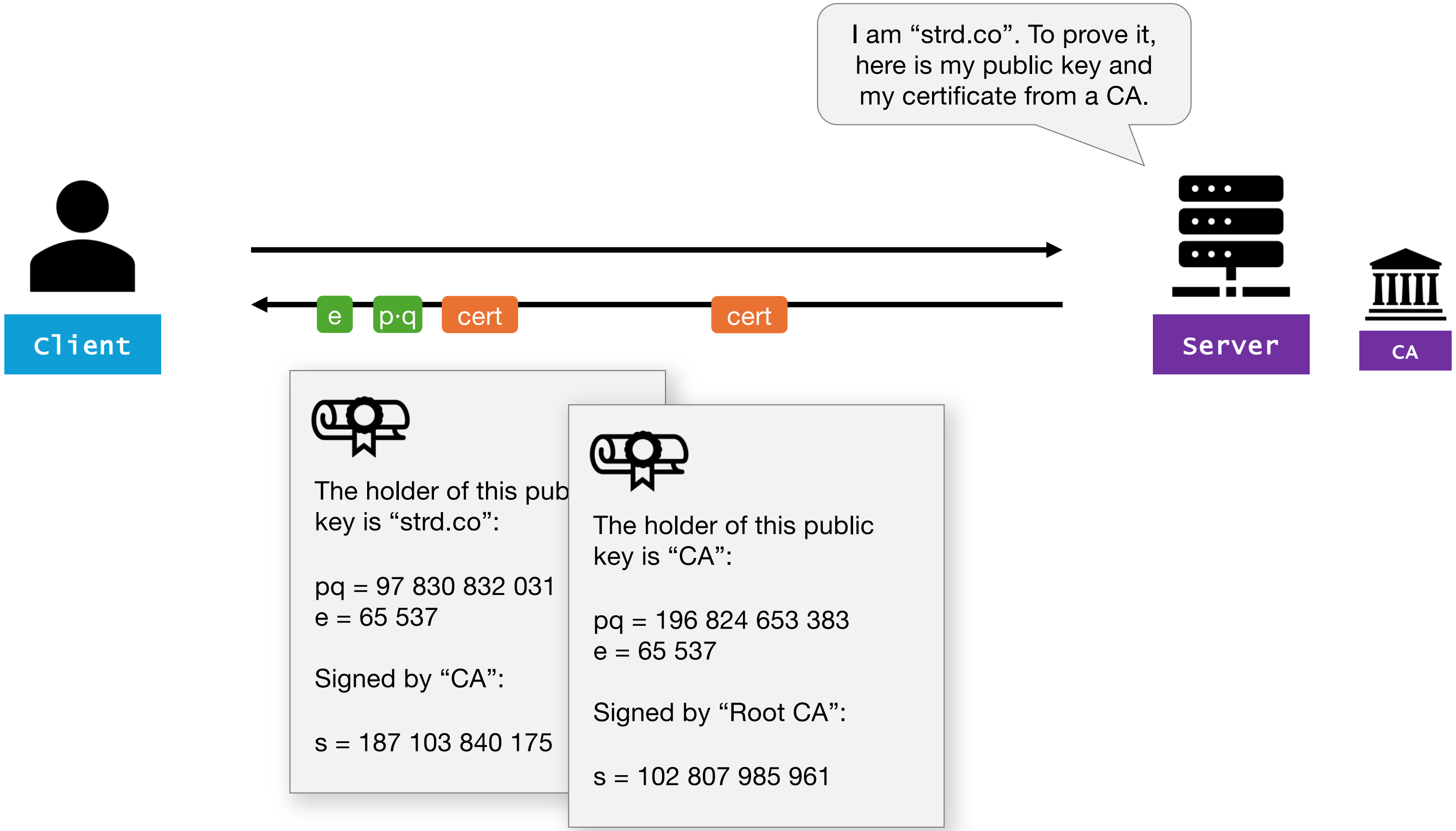
Certificate chains



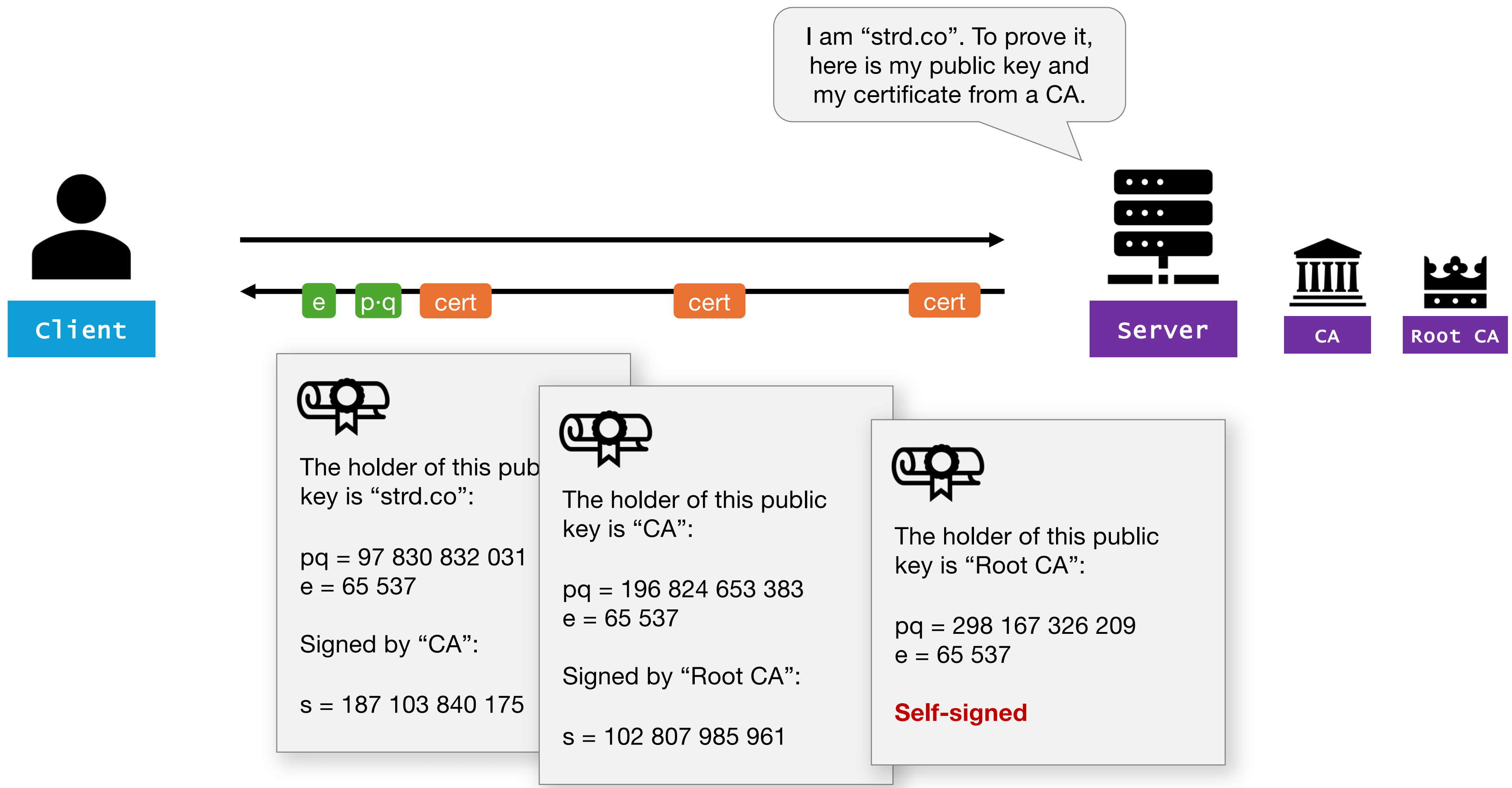
Certificate chains



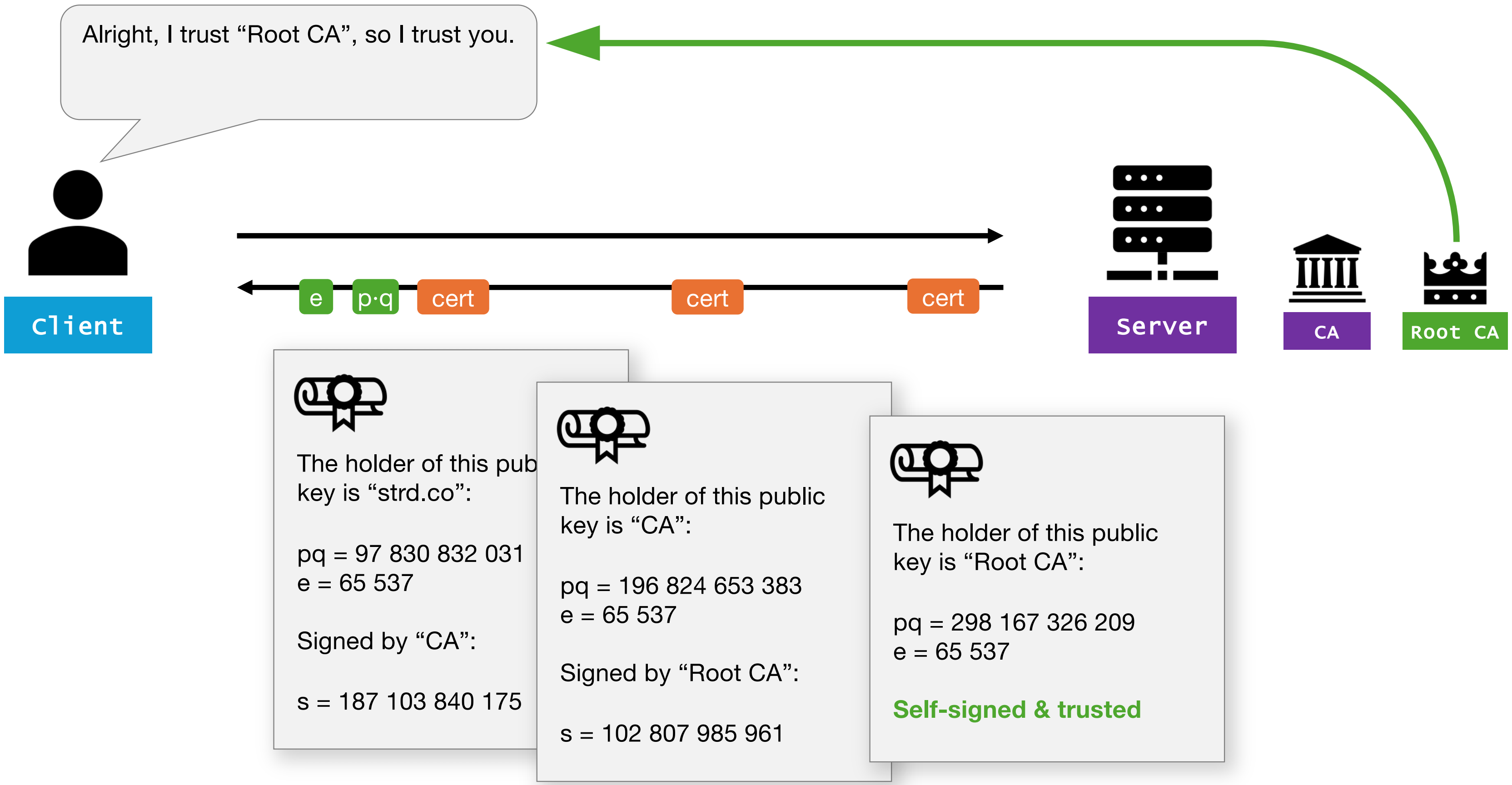
Certificate chains



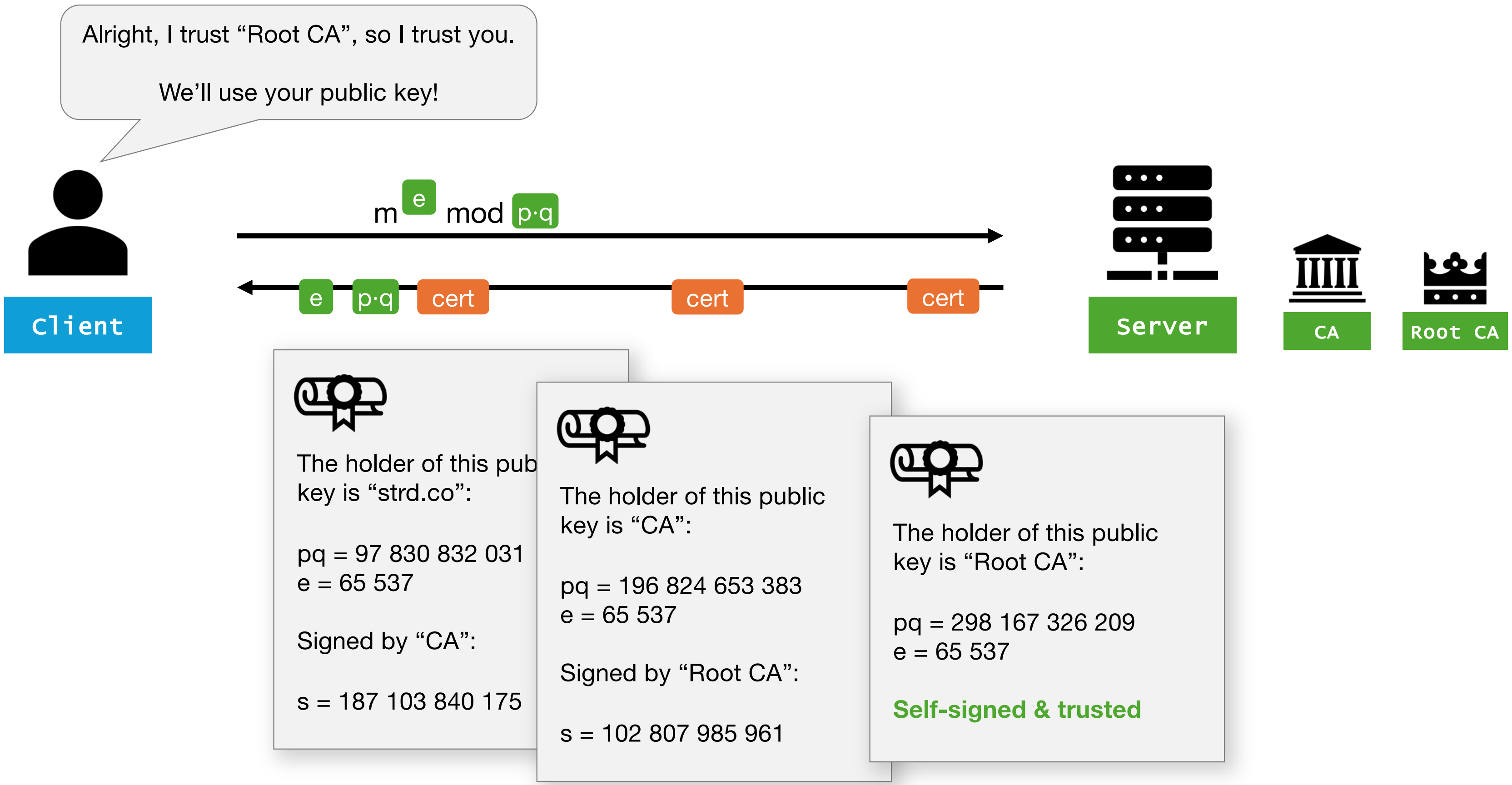
Certificate chains



Certificate chains



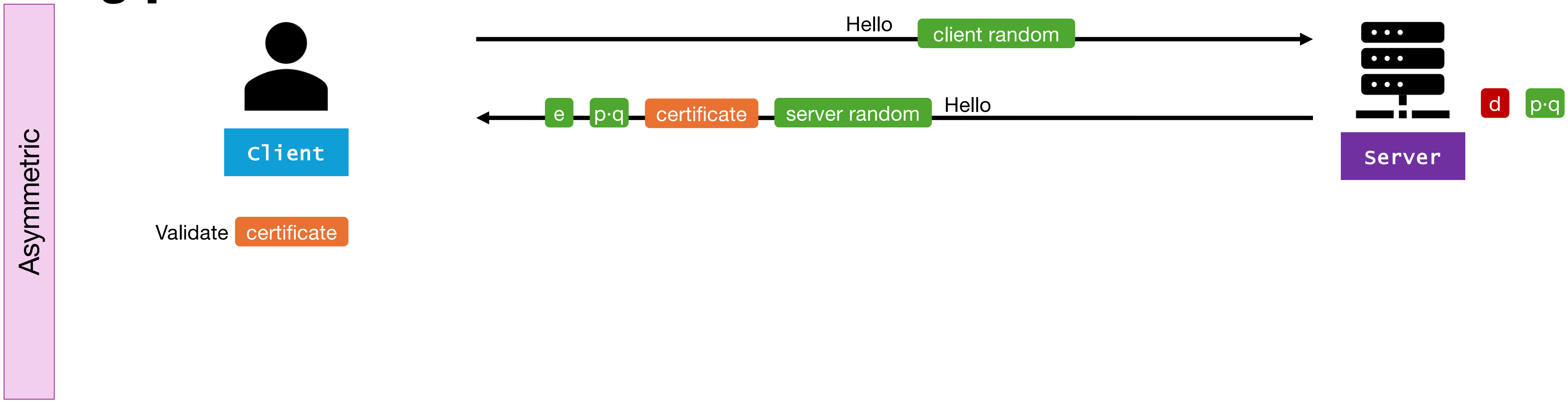
Certificate chains



Certificate chains

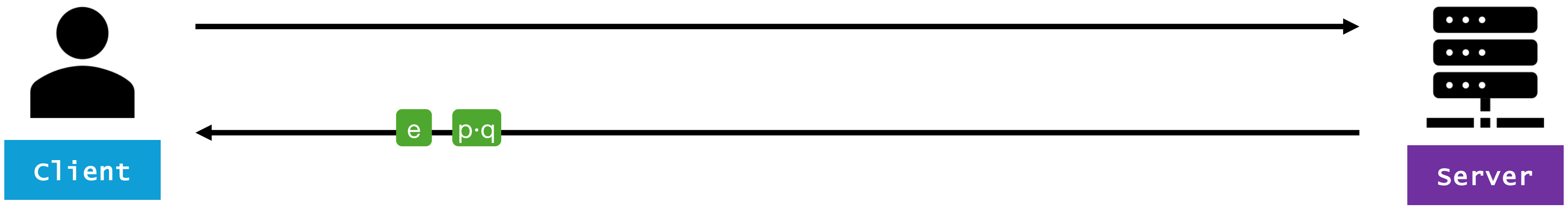
- A certificate is a “digital identity card”. The holder can prove that it has a matching private key using without publishing it.
- Just like identity cards, certificates have authorized issuers.
- Certificates can be self-issued (“self-signed”).
- Certificates mitigates man-in-the-middle (MITM) attacks.

Big picture

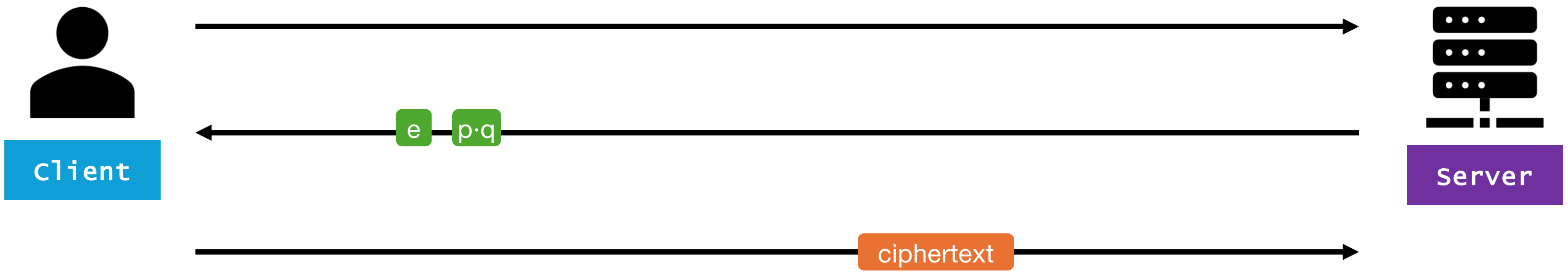

Asymmetric
Client
Validate certificate
Hello
client random
e
p·q
certificate
server random
Hello
d
p·q
Server

MITM

MITM



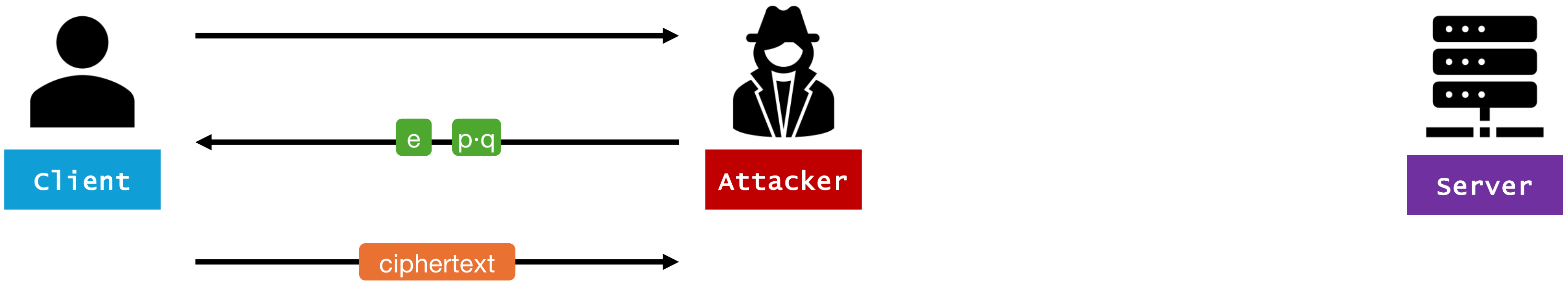
MITM



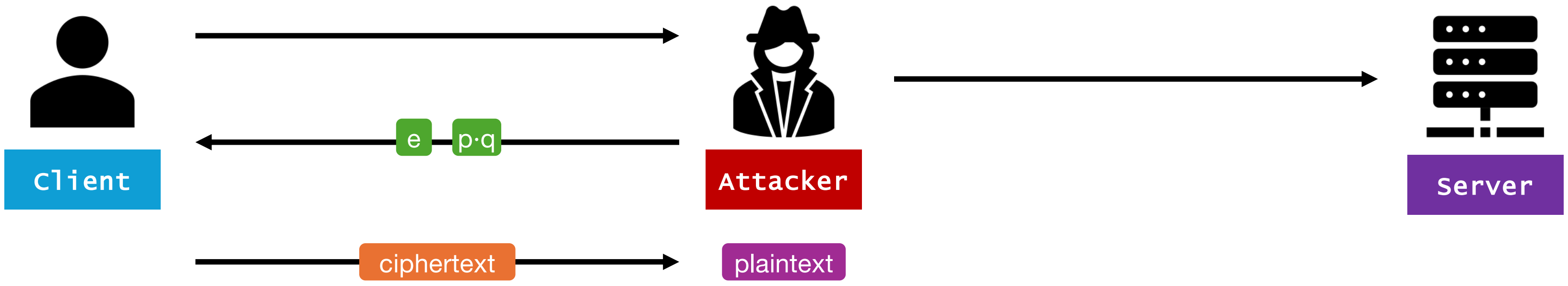
MITM

MITM

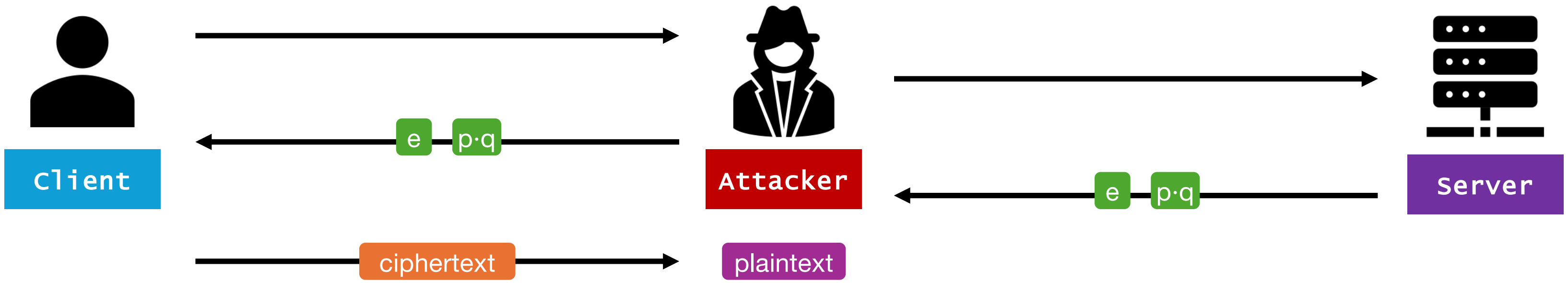
MITM



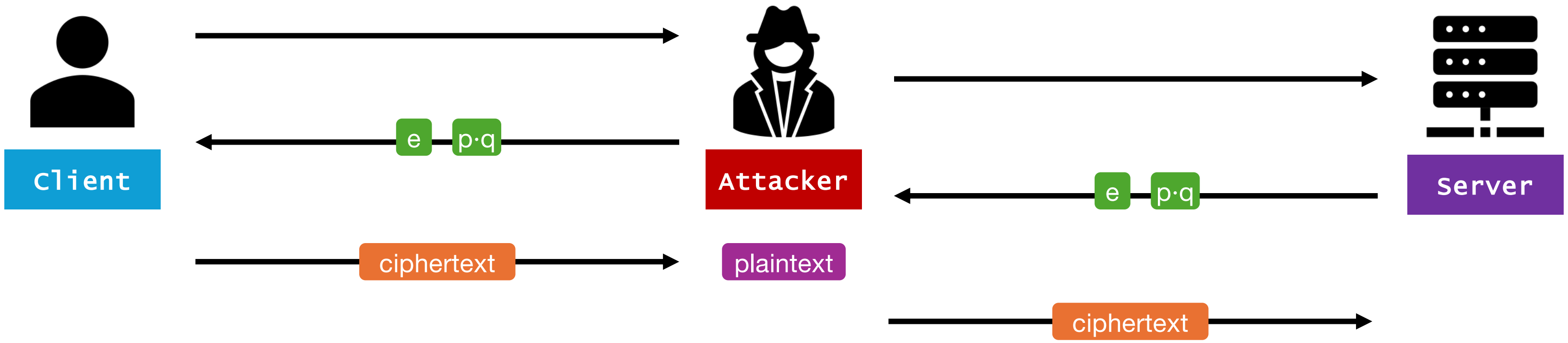
MITM



MITM

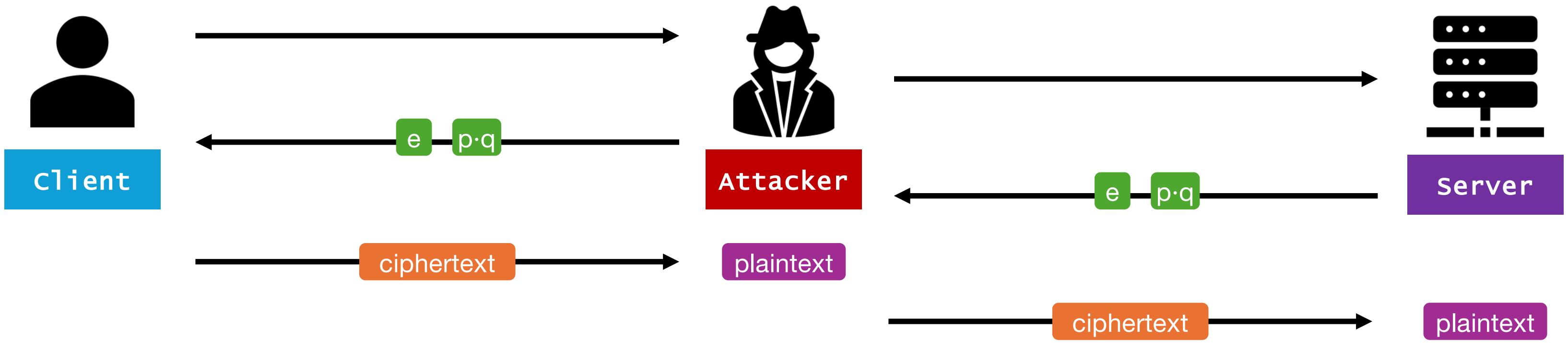


MITM



Client
e
p·q
ciphertext
Attacker
plaintext
e
p·q
ciphertext
Server

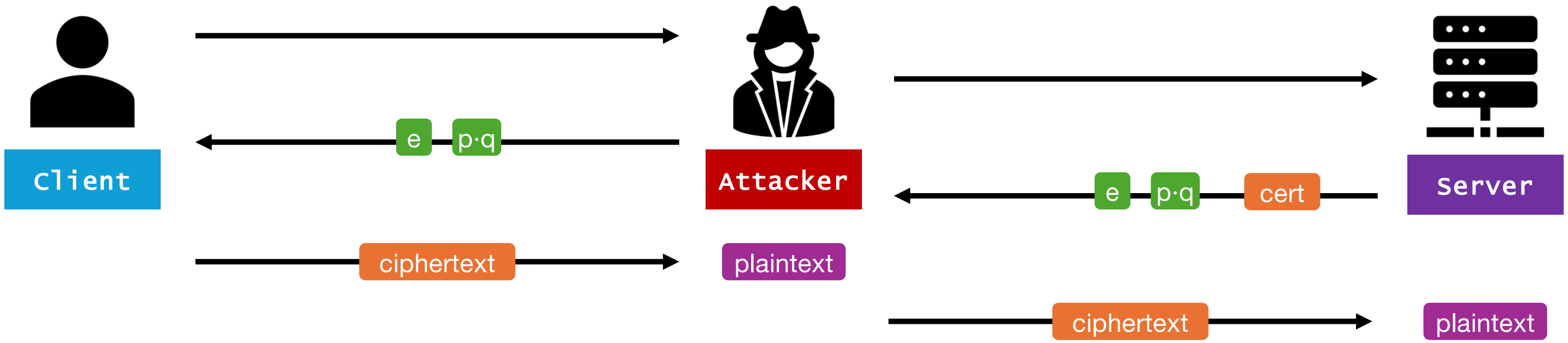
MITM



Client
Attacker
Server
e
p·q
e
p·q
ciphertext
plaintext
ciphertext
plaintext

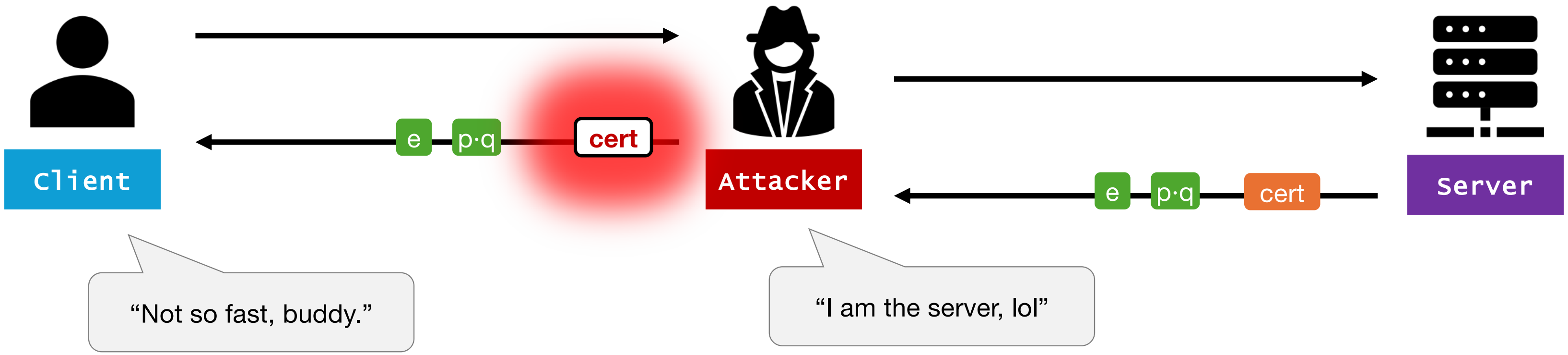
MITM

MITM attacks don't work when the client checks for a certificate:



MITM

MITM attacks don't work when the client checks for a certificate:

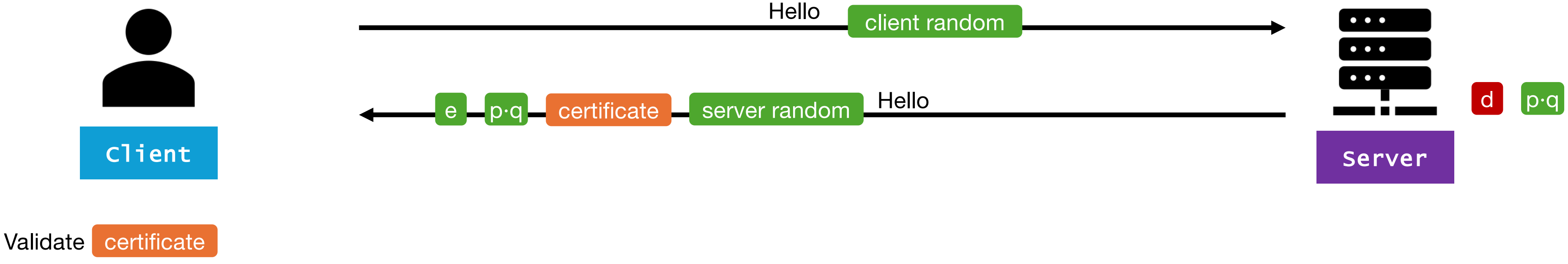


MITM

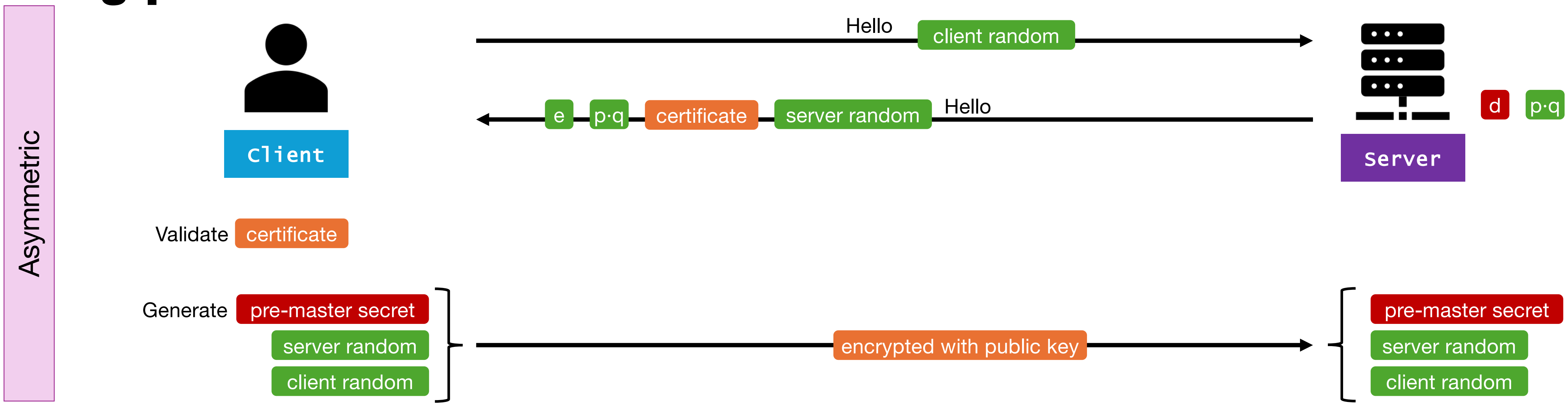
- Your certificate chain is only as strong as its weakest link.
- Be careful with the “trust certificate” checkbox.
- Modern firewalls inspect TLS traffic using a MITM pattern.

Big picture

Asymmetric

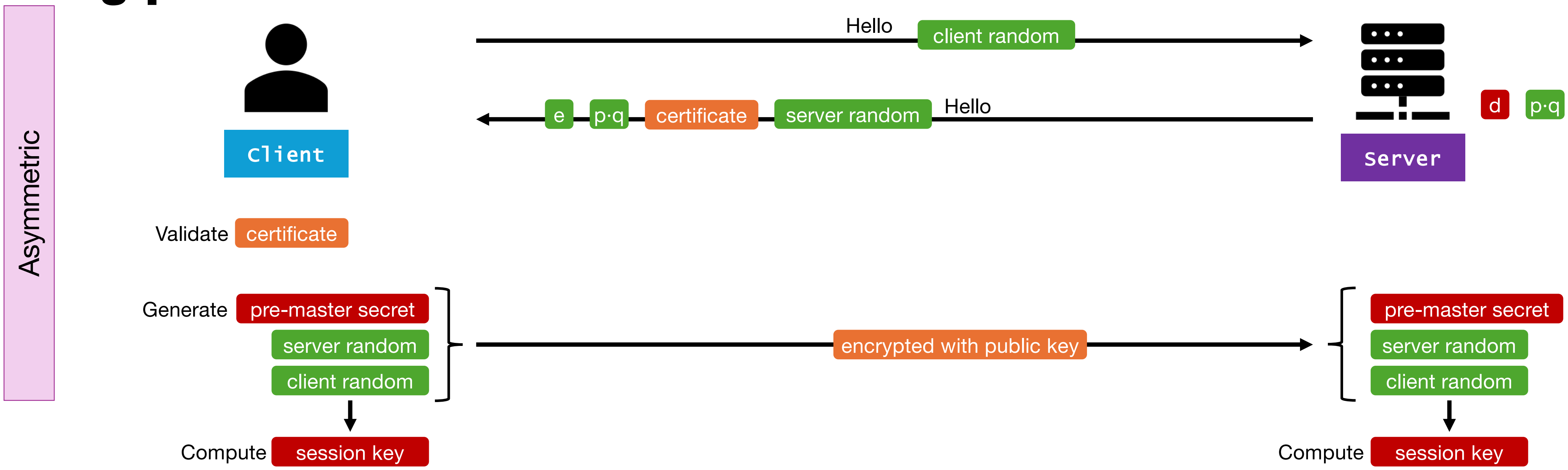


Big picture

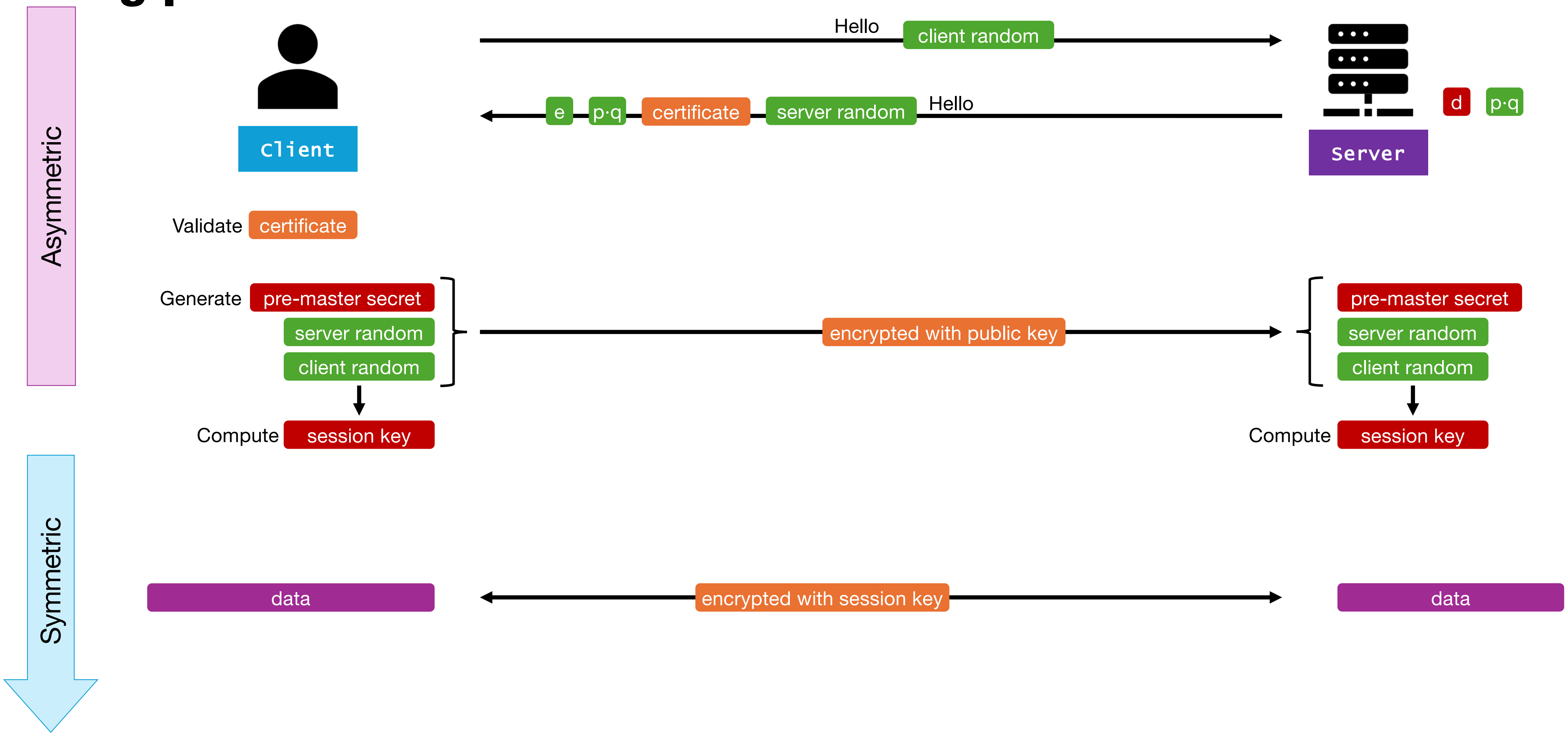


Asymmetric
Client
Hello
client random
Server
d
p·q
e
p·q
certificate
server random
Hello
Validate
certificate
Generate
pre-master secret
server random
client random
encrypted with public key
pre-master secret
server random
client random

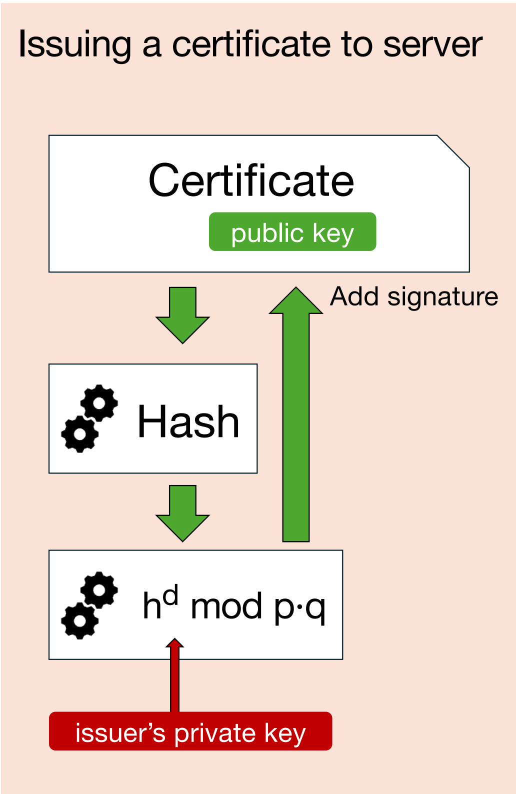
Big picture

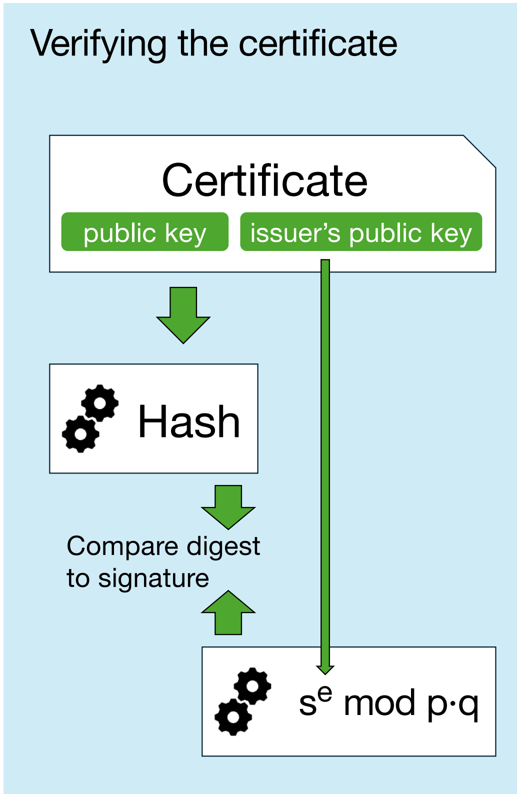

Asymmetric
Client
Hello
client random
e
p·q
certificate
server random
Hello
d
p·q
Server
Validate
certificate
Generate
pre-master secret
server random
client random
encrypted with public key
pre-master secret
server random
client random
Compute
session key
Compute
session key

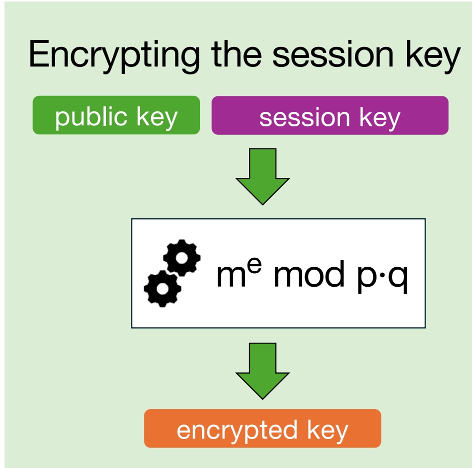
Big picture

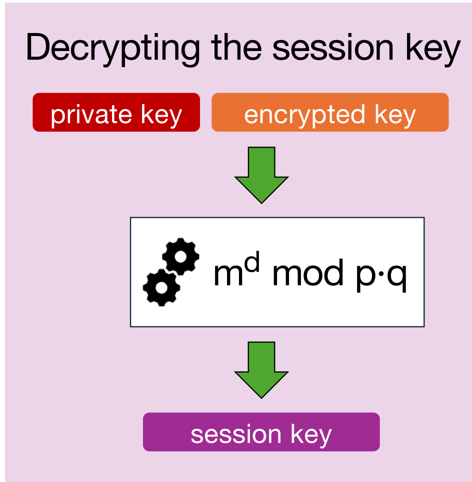

Asymmetric
Symmetric
Client
Server
Hello
client random
e
p·q
certificate
server random
Hello
d
p·q
Validate
certificate
Generate
pre-master secret
server random
client random
encrypted with public key
pre-master secret
server random
client random
Compute
session key
Compute
session key
data
encrypted with session key
data

All the moving parts





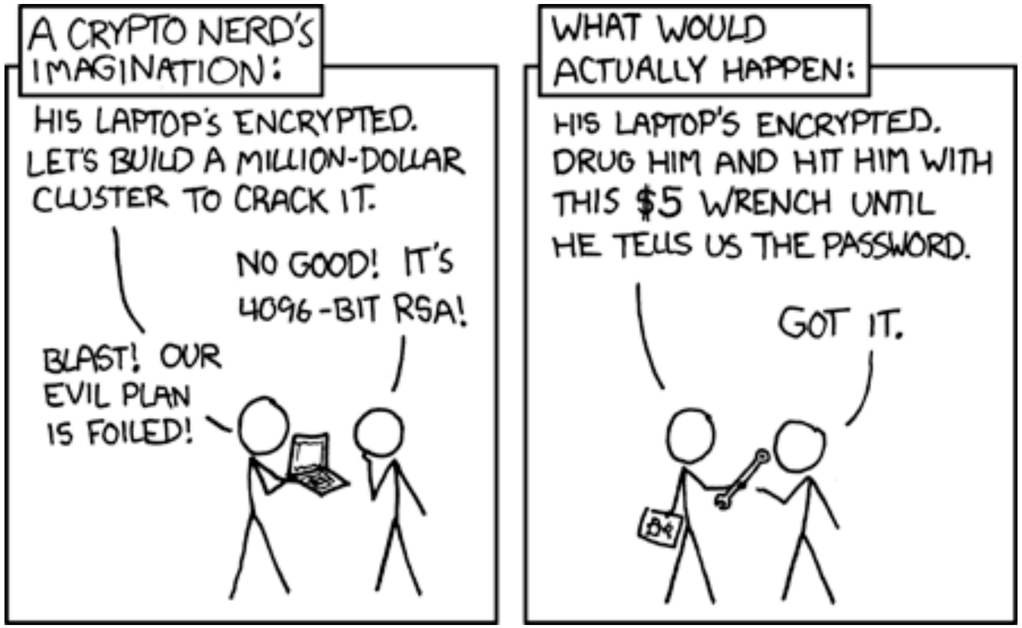




How to defeat strong encryption

How to defeat strong encryption

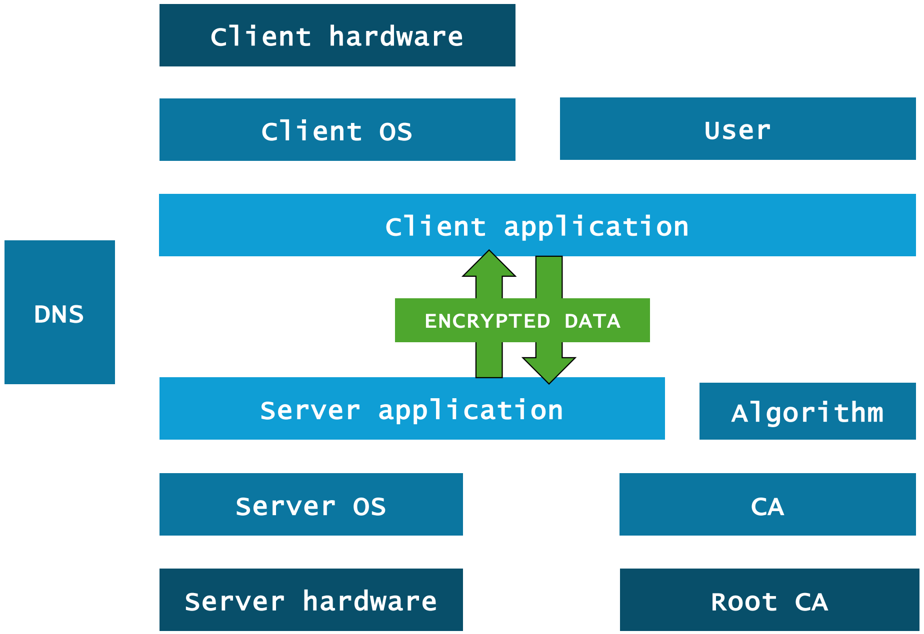
When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.



https://xkcd.com/538/

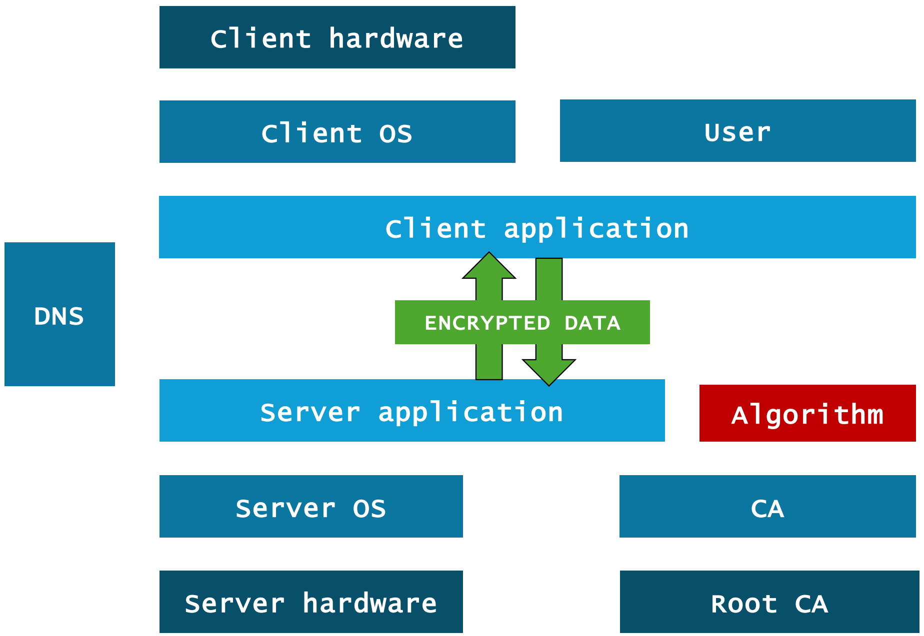
How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.



How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.



2007: Dual_EC_DRBG backdoor found

Reported by Microsoft, the NSA was shown to have introduced an intentional backdoor in a random number algorithm used for ECC. Withdrawn in 2014.

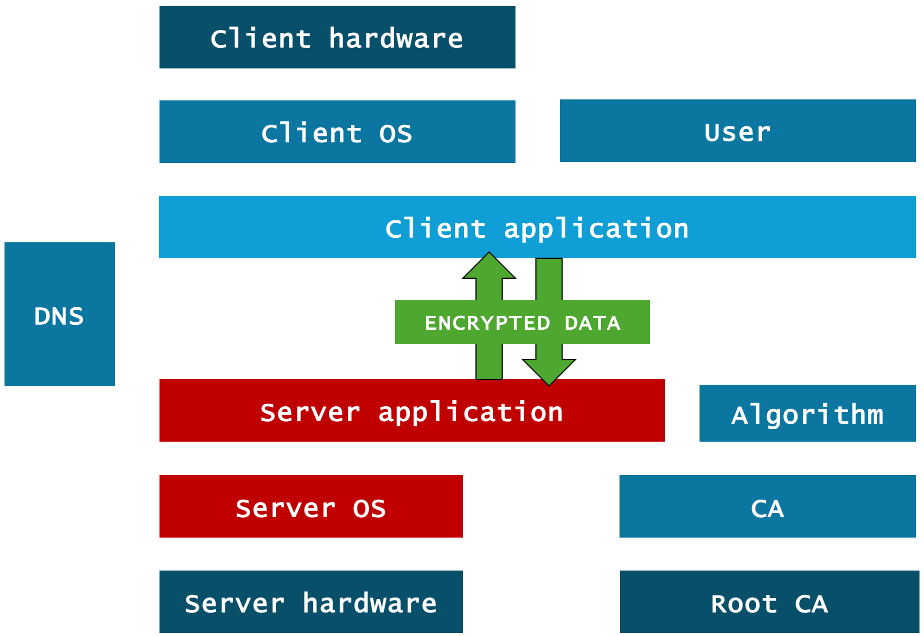
How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.

2020: EncroChat

French police installed a “technical tool” on EncroChat servers, allowing them to covertly log all communications for several months.

See also: Sky ECC

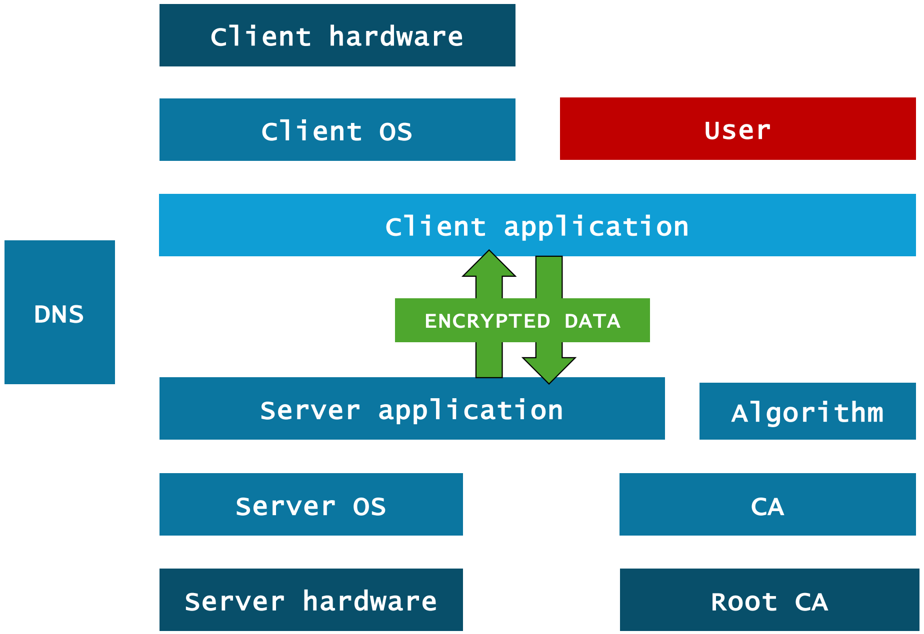


How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.

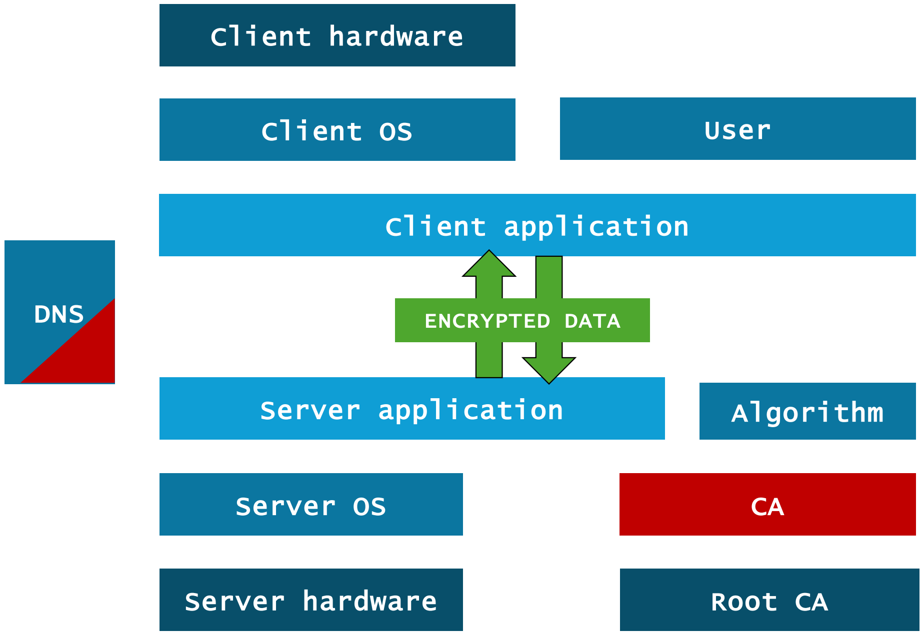
2018-2021: Trojan Shield

In an elaborate honey pot/trojan attack, law enforcement agencies distributed ANOM, a “secure chat” application, which they then could monitor.



How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.



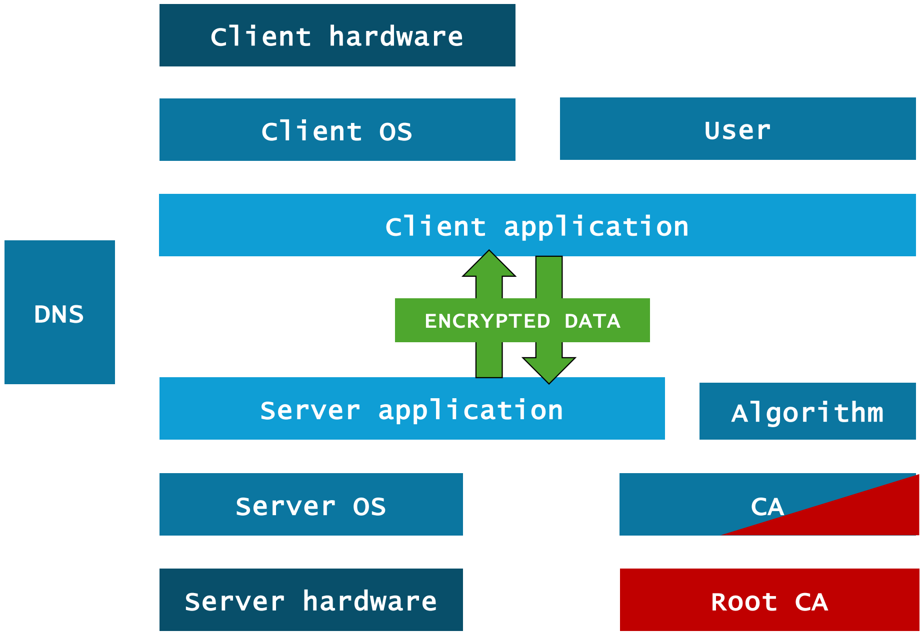
2011: DigiNotar

Dutch CA DigiNotar was hacked and used to issue fraudulent certificates for Gmail.

The certificates were likely used to spy on 300 000 Iranian users' accounts using a MITM attack.

How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.



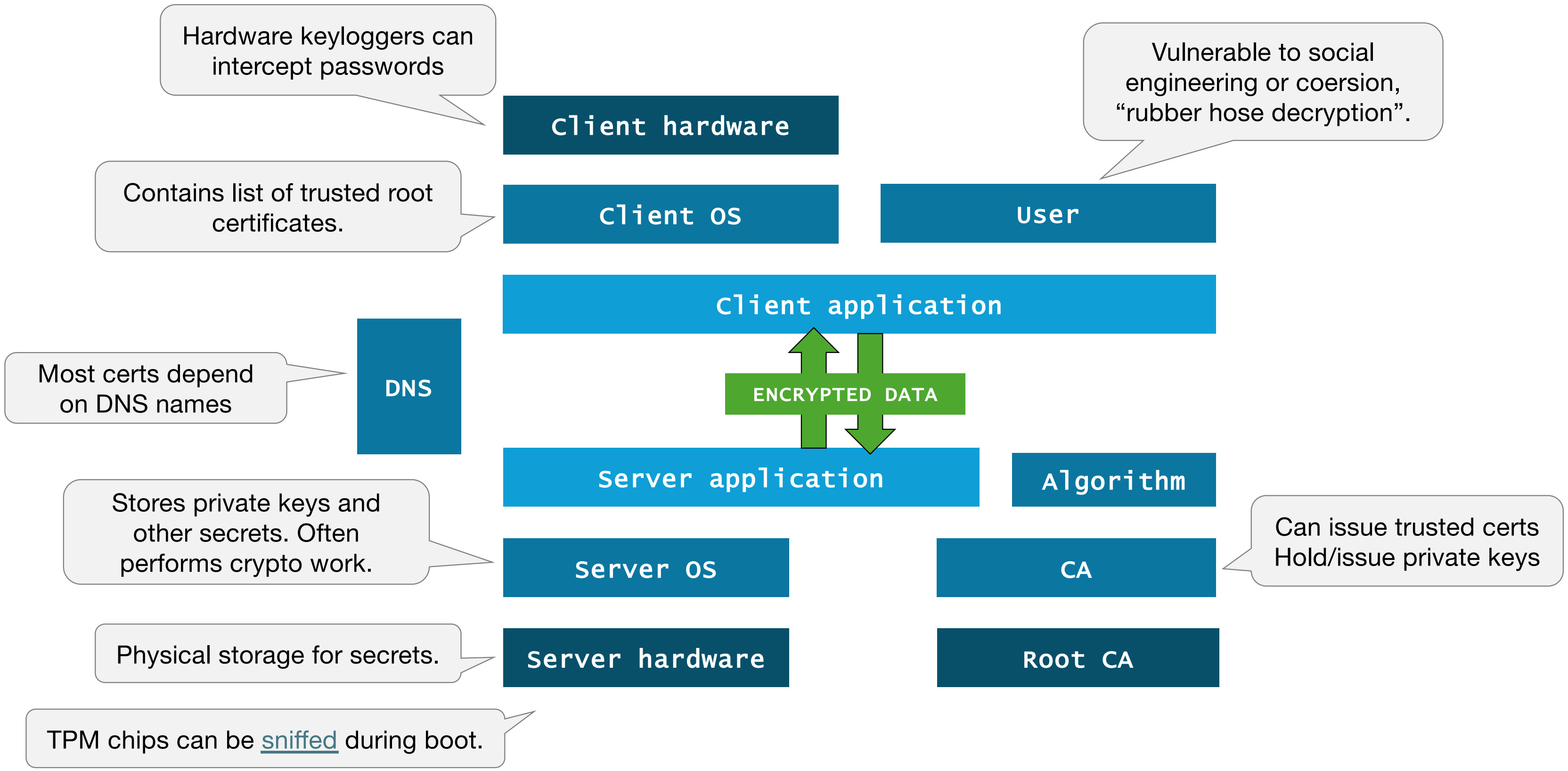
2019: Kazakhstan

Starting in July 2019, the Kazakh government started instructing citizens to install a root certificate.

Major browser developers have for years refused to add the root cert, and proceeded to block the cert.

How to defeat strong encryption

When a cryptographical attack is not feasible, an opponent could try to bypass the encryption entirely.



Mmmkay

github.com/sqlsunday/presentations

Feel free to reach out with questions or comments.

Email: daniel@strd.co
Blog: sqlsunday.com
Bluesky: @dhma.ch

Microsoft Most Valuable Professional

Tools, links, docs

AES:

- AES visualizer: http://advancedcomputing.org/aes_visualizer.html

RSA:

- Interactive RSA calculation: https://www.henryschmale.org/2022/03/14/rsa.html
- RSA numbers: https://en.wikipedia.org/wiki/RSA_numbers
- Big number calculator: https://defuse.ca/big-number-calculator.htm
- Prime factorization tool: http://www.javascripter.net/math/calculators/primefactorscalculator.htm
- Fermat attack on RSA (when p and q are close to the square root): https://fermatattack.secvuln.info/