CS250: Discrete Math for Computer Science

L27: Cryptography and RSA

Recall: Fermat's Little Theorem

Thm: For p prime, $a \in \mathbf{Z}_p^\star$, $a^{p-1} \equiv 1 \pmod{p}$

Recall: Fermat's Little Theorem

Thm: For p prime, $a \in \mathbf{Z}_p^\star$, $a^{p-1} \equiv 1 \pmod{p}$

Proof: $f_a : \mathbf{Z}_p^\star \underset{\text{onto}}{\overset{1:1}{\rightarrow}} \mathbf{Z}_p^\star$

Recall: Fermat's Little Theorem

Thm: For p prime, $\quad a \in \mathbf{Z}_p^\star, \quad a^{p-1} \equiv 1 \pmod p$

Proof: $\quad f_a : \mathbf{Z}_p^\star \underset{\text{onto}}{\overset{1:1}{\to}} \mathbf{Z}_p^\star$

$$f_a(x) = (a \cdot x) \qquad f_a^{-1}(x) = ((a^{-1}(\bmod p)) \cdot x)$$

Recall: Fermat's Little Theorem

Thm: For p prime, $a \in \mathbf{Z}_p^\star$, $a^{p-1} \equiv 1 \pmod{p}$

Proof: $f_a : \mathbf{Z}_p^\star \underset{\text{onto}}{\overset{1:1}{\to}} \mathbf{Z}_p^\star$

$$f_a(x) = (a \cdot x) \qquad f_a^{-1}(x) = ((a^{-1}(\bmod p)) \cdot x)$$

$$Z_p^\star = \{1, 2, \ldots, p-1\} = \{f_a(1), f_a(2), \ldots, f_a(p-1)\}$$

Recall: Fermat's Little Theorem

Thm: For p prime, $a \in \mathbf{Z}_p^\star$, $a^{p-1} \equiv 1 \pmod{p}$

Proof: $f_a : \mathbf{Z}_p^\star \underset{\text{onto}}{\overset{1:1}{\rightarrow}} \mathbf{Z}_p^\star$

$$f_a(x) = (a \cdot x) \qquad f_a^{-1}(x) = ((a^{-1}(\bmod p)) \cdot x)$$

$$Z_p^\star = \{1, 2, \ldots, p-1\} = \{f_a(1), f_a(2), \ldots, f_a(p-1)\}$$

$$\{1, 2, \ldots, p-1\} = \{a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)\}$$

Recall: Fermat's Little Theorem

Thm: For p prime, $\quad a \in \mathbf{Z}_p^\star, \quad a^{p-1} \equiv 1 \pmod p$

Proof: $\quad f_a : \mathbf{Z}_p^\star \xrightarrow[\text{onto}]{1:1} \mathbf{Z}_p^\star$

$$f_a(x) = (a \cdot x) \qquad f_a^{-1}(x) = ((a^{-1}(\bmod p)) \cdot x)$$

$$Z_p^\star = \{1, 2, \ldots, p-1\} = \{f_a(1), f_a(2), \ldots, f_a(p-1)\}$$

$$\{1, 2, \ldots, p-1\} = \{a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)\}$$

$$\prod_{i \in \mathbf{Z}_p^\star} i \equiv \prod_{i \in \mathbf{Z}_p^\star} a \cdot i \pmod p$$

Recall: Fermat's Little Theorem

Thm: For p prime, $\quad a \in \mathbf{Z}_p^\star, \quad a^{p-1} \equiv 1 \pmod p$

Proof: $\quad f_a : \mathbf{Z}_p^\star \overset{1:1}{\underset{\text{onto}}{\rightarrow}} \mathbf{Z}_p^\star$

$$f_a(x) = (a \cdot x) \qquad f_a^{-1}(x) = ((a^{-1}(\bmod p)) \cdot x)$$

$$Z_p^\star = \{1, 2, \ldots, p-1\} = \{f_a(1), f_a(2), \ldots, f_a(p-1)\}$$

$$\{1, 2, \ldots, p-1\} = \{a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)\}$$

$$\prod_{i \in \mathbf{Z}_p^\star} i \equiv \prod_{i \in \mathbf{Z}_p^\star} a \cdot i \pmod p$$

$$\prod_{i \in \mathbf{Z}_p^\star} i \equiv a^{p-1} \prod_{i \in \mathbf{Z}_p^\star} i \pmod p$$

Recall: Fermat's Little Theorem

Thm: For p prime, $a \in \mathbf{Z}_p^\star$, $a^{p-1} \equiv 1 \pmod p$

Proof: $f_a : \mathbf{Z}_p^\star \underset{\text{onto}}{\overset{1:1}{\to}} \mathbf{Z}_p^\star$

$$f_a(x) = (a \cdot x) \qquad f_a^{-1}(x) = ((a^{-1} (\bmod p)) \cdot x)$$

$$Z_p^\star = \{1, 2, \ldots, p-1\} = \{f_a(1), f_a(2), \ldots, f_a(p-1)\}$$

$$\{1, 2, \ldots, p-1\} = \{a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)\}$$

$$\prod_{i \in \mathbf{Z}_p^\star} i \equiv \prod_{i \in \mathbf{Z}_p^\star} a \cdot i \pmod p$$

$$\prod_{i \in \mathbf{Z}_p^\star} i \equiv a^{p-1} \prod_{i \in \mathbf{Z}_p^\star} i \pmod p$$

$$1 \equiv a^{p-1} \pmod p \qquad \square$$

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\varphi(p) = p - 1$$

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\begin{aligned} \varphi(p) &= p-1 \\ \varphi(p^{k+1}) &= (p-1)p^k \end{aligned}$$

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\begin{aligned} \varphi(p) &= p - 1 \\ \varphi(p^{k+1}) &= (p-1)p^k \end{aligned}$$

If gcd$(a, b) = 1$**,**

$$\varphi(ab) = \varphi(a)\varphi(b)$$

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\begin{aligned} \varphi(p) &= p-1 \\ \varphi(p^{k+1}) &= (p-1)p^k \end{aligned}$$

If $\gcd(a, b) = 1$**,**

$$\varphi(ab) = \varphi(a)\varphi(b)$$

Why?

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\begin{aligned} \varphi(p) &= p-1 \\ \varphi(p^{k+1}) &= (p-1)p^k \end{aligned}$$

If $\gcd(a,b) = 1$**,**

$$\varphi(ab) = \varphi(a)\varphi(b)$$

Why?

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\begin{aligned} \varphi(p) &= p-1 \\ \varphi(p^{k+1}) &= (p-1)p^k \end{aligned}$$

If $\gcd(a,b)=1$**,**

$$\varphi(ab) = \varphi(a)\varphi(b)$$

Why? **CRT, hw5**

Euler's phi function, $\varphi(n) = |\mathbf{Z}_n^*|$

n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$	n	$\varphi(\mathbf{n})$
2	**1**	11	**10**	20	**8**
3	**2**	12	**4**	21	**12**
4	**2**	13	**12**	22	**10**
5	**4**	14	**6**	23	**22**
6	**2**	15	**8**	24	**8**
7	**6**	16	**8**	25	**20**
8	**4**	17	**16**	26	**12**
9	**6**	18	**6**	27	**18**
10	**4**	19	**18**	28	**12**

What's the pattern?

For p prime,

$$\varphi(p) = p - 1$$

$$\varphi(p^{k+1}) = (p-1)p^k$$

If $\gcd(a, b) = 1$,

$$\varphi(ab) = \varphi(a)\varphi(b)$$

Why? CRT, hw5

For primes, $p \neq q$,

$$\varphi(pq) = (p-1)(q-1)$$

Euler's Thm:

For $m > 1$, $\quad a \in \mathbf{Z}_m^\star, \quad a^{\varphi(m)} \equiv 1 \pmod{m}$.

Euler's Thm:

For $m > 1$, $\quad a \in \mathbf{Z}_m^\star, \quad a^{\varphi(m)} \equiv 1 \pmod{m}$.

proof: For $a \in \mathbf{Z}_m^\star$, $\quad f_a : \mathbf{Z}_m^\star \underset{\text{onto}}{\overset{1:1}{\rightarrow}} \mathbf{Z}_m^\star, \quad f_a(x) = (a \cdot x) \,\%\, m$

Euler's Thm:

For $m > 1$, $\quad a \in \mathbf{Z}_m^\star, \quad a^{\varphi(m)} \equiv 1 \pmod{m}$.

proof: For $a \in \mathbf{Z}_m^\star, \quad f_a : \mathbf{Z}_m^\star \overset{1:1}{\underset{\text{onto}}{\rightarrow}} \mathbf{Z}_m^\star, \quad f_a(x) = (a \cdot x) \,\%\, m$

$$Z_m^\star \;=\; \{b_1, \ldots, b_{\varphi(m)}\} \quad = \quad \{f_a(b_1), \ldots, f_a(b_{\varphi(m)})\}$$

Euler's Thm:

For $m > 1$, $\quad a \in \mathbf{Z}_m^\star$, $\quad a^{\varphi(m)} \equiv 1 \pmod{m}$.

proof: For $a \in \mathbf{Z}_m^\star$, $\quad f_a : \mathbf{Z}_m^\star \overset{1:1}{\underset{\text{onto}}{\rightarrow}} \mathbf{Z}_m^\star$, $\quad f_a(x) = (a \cdot x) \,\%\, m$

$$Z_m^\star \; = \; \{b_1, \ldots, b_{\varphi(m)}\} \quad = \quad \{f_a(b_1), \ldots, f_a(b_{\varphi(m)})\}$$

$$\{b_1 \ldots, b_{\varphi(m)}\} \quad = \quad \{a \cdot b_1, \ldots, a \cdot b_{\varphi(m)}\}$$

Euler's Thm:

For $m > 1$, $\quad a \in \mathbf{Z}_m^\star, \quad a^{\varphi(m)} \equiv 1 \pmod{m}$.

proof: For $a \in \mathbf{Z}_m^\star, \quad f_a : \mathbf{Z}_m^\star \overset{1:1}{\underset{\text{onto}}{\rightarrow}} \mathbf{Z}_m^\star, \quad f_a(x) = (a \cdot x) \,\%\, m$

$$Z_m^\star \;=\; \{b_1, \ldots, b_{\varphi(m)}\} \quad = \quad \{f_a(b_1), \ldots, f_a(b_{\varphi(m)})\}$$

$$\{b_1 \ldots, b_{\varphi(m)}\} \quad = \quad \{a \cdot b_1, \ldots, a \cdot b_{\varphi(m)}\}$$

$$\prod_{b \in \mathbf{Z}_m^\star} b \quad \equiv \quad \prod_{b \in \mathbf{Z}_m^\star} a \cdot b \quad (\text{mod } m)$$

Euler's Thm:

For $m > 1$, $\quad a \in \mathbf{Z}_m^\star$, $\quad a^{\varphi(m)} \equiv 1 \pmod{m}$.

proof: For $a \in \mathbf{Z}_m^\star$, $\quad f_a : \mathbf{Z}_m^\star \underset{\text{onto}}{\overset{1:1}{\rightarrow}} \mathbf{Z}_m^\star$, $\quad f_a(x) = (a \cdot x) \,\%\, m$

$$Z_m^\star = \{b_1, \ldots, b_{\varphi(m)}\} \quad = \quad \{f_a(b_1), \ldots, f_a(b_{\varphi(m)})\}$$

$$\{b_1 \ldots, b_{\varphi(m)}\} \quad = \quad \{a \cdot b_1, \ldots, a \cdot b_{\varphi(m)}\}$$

$$\prod_{b \in \mathbf{Z}_m^\star} b \equiv \prod_{b \in \mathbf{Z}_m^\star} a \cdot b \pmod{m}$$

$$\prod_{b \in \mathbf{Z}_m^\star} b \equiv a^{\varphi(m)} \prod_{b \in \mathbf{Z}_p^\star} b \pmod{m}$$

Euler's Thm:

For $m > 1$, $a \in \mathbf{Z}_m^\star$, $a^{\varphi(m)} \equiv 1 \pmod{m}$.

proof: For $a \in \mathbf{Z}_m^\star$, $f_a : \mathbf{Z}_m^\star \underset{\text{onto}}{\overset{1:1}{\rightarrow}} \mathbf{Z}_m^\star$, $f_a(x) = (a \cdot x) \,\%\, m$

$$Z_m^\star = \{b_1, \ldots, b_{\varphi(m)}\} = \{f_a(b_1), \ldots, f_a(b_{\varphi(m)})\}$$

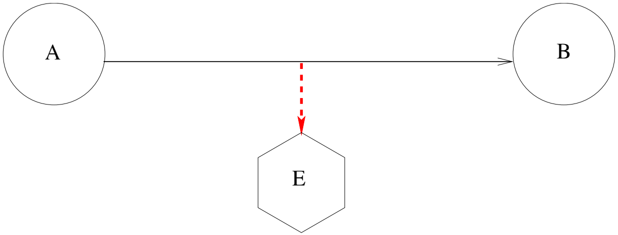
$$\{b_1 \ldots, b_{\varphi(m)}\} = \{a \cdot b_1, \ldots, a \cdot b_{\varphi(m)}\}$$

$$\prod_{b \in \mathbf{Z}_m^\star} b \equiv \prod_{b \in \mathbf{Z}_m^\star} a \cdot b \pmod{m}$$

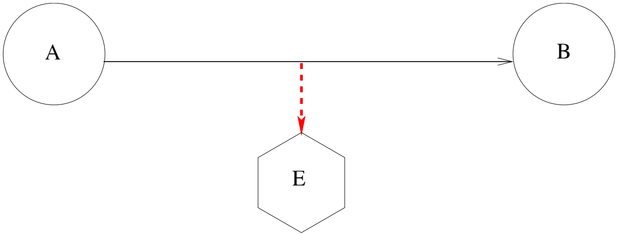
$$\prod_{b \in \mathbf{Z}_m^\star} b \equiv a^{\varphi(m)} \prod_{b \in \mathbf{Z}_p^\star} b \pmod{m}$$

$$1 \equiv a^{\varphi(m)} \pmod{m} \qquad \square$$

Cryptography

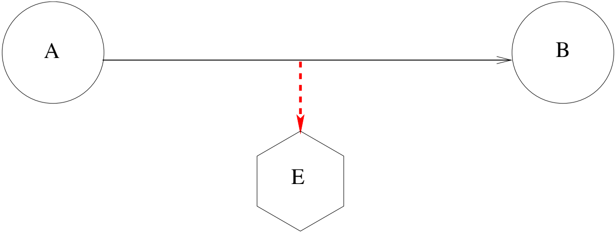


Cryptography



One-Time Pad: a perfectly secure cryptosystem

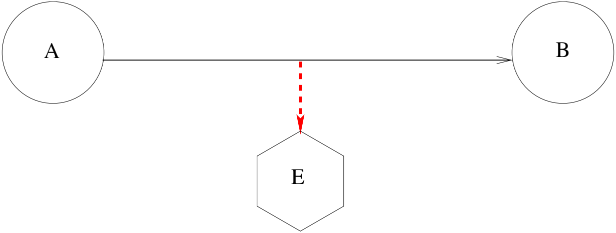
Cryptography



One-Time Pad: a perfectly secure cryptosystem

$p \in \{0,1\}^n$ $\quad m \in \{0,1\}^n \quad = \quad$ binary strings of length n

Cryptography

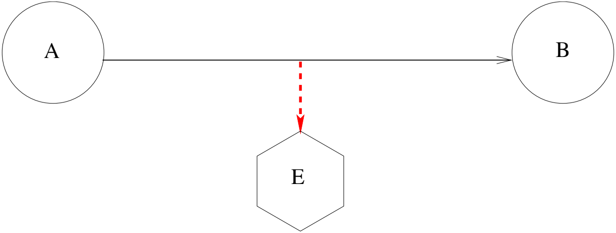


One-Time Pad: a perfectly secure cryptosystem

$$p \in \{0,1\}^n \qquad m \in \{0,1\}^n = \text{binary strings of length } n$$

$$E(p,x) = p \oplus x$$

Cryptography



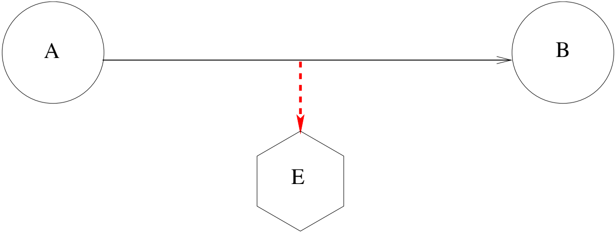
One-Time Pad: a perfectly secure cryptosystem

$p \in \{0,1\}^n \qquad m \in \{0,1\}^n \quad = \quad$ binary strings of length n

$$E(p,x) = p \oplus x$$

$$D(p,x) = p \oplus x$$

Cryptography

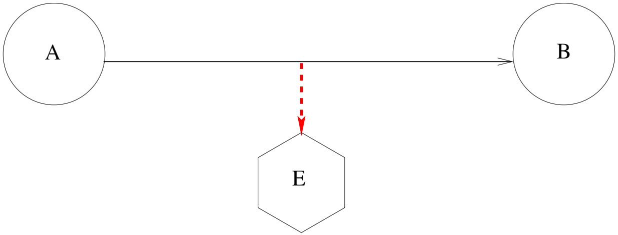


One-Time Pad: a perfectly secure cryptosystem

$p \in \{0,1\}^n \qquad m \in \{0,1\}^n \quad = \quad$ binary strings of length n

$$
\begin{aligned}
E(p,x) &= p \oplus x \\
D(p,x) &= p \oplus x \\
D(p,E(p,m)) &= p \oplus (p \oplus m) \quad = \quad m
\end{aligned}
$$

Cryptography



One-Time Pad: a perfectly secure cryptosystem

$$p \in \{0,1\}^n \qquad m \in \{0,1\}^n \quad = \quad \text{binary strings of length } n$$

$$E(p,x) = p \oplus x$$

$$D(p,x) = p \oplus x$$

$$D(p, E(p,m)) = p \oplus (p \oplus m) = m$$

Encryption and **decryption** functions are the same:
bitwise **exclusive or** with **random**, **secret** one-time pad, p.

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1

$$E(p, m) \ = \ p \oplus m \qquad\qquad D(p, s) \ = \ p \oplus s$$

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0

$$E(p, m) = p \oplus m \qquad D(p, s) = p \oplus s$$

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0
$E(p, m)$	0	1	1	0	1	0	1	0	0	1

$$E(p, m) = p \oplus m \qquad D(p, s) = p \oplus s$$

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0
$E(p, m)$	0	1	1	0	1	0	1	0	0	1
$D(p, E(p, m))$	0	0	0	0	1	1	1	1	0	0

$$E(p, m) = p \oplus m \qquad D(p, s) = p \oplus s$$

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0
$E(p, m)$	0	1	1	0	1	0	1	0	0	1
$D(p, E(p, m))$	0	0	0	0	1	1	1	1	0	0

$$E(p, m) = p \oplus m \qquad D(p, s) = p \oplus s$$

Thm: If p is **chosen at random** and **known only** by A and B,

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0
$E(p,m)$	0	1	1	0	1	0	1	0	0	1
$D(p,E(p,m))$	0	0	0	0	1	1	1	1	0	0

$$E(p,m) \;=\; p \oplus m \qquad\qquad D(p,s) \;=\; p \oplus s$$

Thm: If p is **chosen at random** and **known only** by A and B,

Then $E(p,m)$ provides **no information** about m

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0
$E(p, m)$	0	1	1	0	1	0	1	0	0	1
$D(p, E(p, m))$	0	0	0	0	1	1	1	1	0	0

$$E(p, m) = p \oplus m \qquad\qquad D(p, s) = p \oplus s$$

Thm: If p is **chosen at random** and **known only** by A and B,

Then $E(p, m)$ provides **no information** about m

except perhaps its length.

One-Time Pad, Continued

p	0	1	1	0	0	1	0	1	0	1
m	0	0	0	0	1	1	1	1	0	0
$E(p,m)$	0	1	1	0	1	0	1	0	0	1
$D(p,E(p,m))$	0	0	0	0	1	1	1	1	0	0

$$E(p,m) = p \oplus m \qquad D(p,s) = p \oplus s$$

Thm: If p is **chosen at random** and **known only** by A and B,

Then $E(p,m)$ provides **no information** about m

except perhaps its length.

Do not use p more than once!

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**,

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me,

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

RSA [Rivest, Shamir, Adleman, 1976]

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

RSA [Rivest, Shamir, Adleman, 1976]

For slightly over 3 weeks, **each day** Rivest and Shamir came up with a new scheme to do public-key cryptography, . . . , and **by the next morning** Adleman had broken it.

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

RSA [Rivest, Shamir, Adleman, 1976]

For slightly over 3 weeks, **each day** Rivest and Shamir came up with a new scheme to do public-key cryptography, . . . , and **by the next morning** Adleman had broken it.
The 23rd scheme, **Adleman couldn't break**.

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

RSA [Rivest, Shamir, Adleman, 1976]

For slightly over 3 weeks, each day Rivest and Shamir came up with a new scheme to do public-key cryptography, . . . , and **by the next morning** Adleman had broken it.
The 23rd scheme, **Adleman couldn't break**.

This is the **RSA Public-Key Algorithm** that is used today in the **SSL algorithm**

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

RSA [Rivest, Shamir, Adleman, 1976]

For slightly over 3 weeks, each day Rivest and Shamir came up with a new scheme to do public-key cryptography, . . . , and **by the next morning** Adleman had broken it.
The 23rd scheme, **Adleman couldn't break**.

This is the **RSA Public-Key Algorithm** that is used today in the **SSL algorithm**

Lets your browser **generate key** to send order to Amazon

Public-Key Cryptography

[Diffie, Hellman, 1976] Using **computational complexity**, **publish key** for sending secret messages to me, **intractable** for anyone but me **to decode**.

RSA [Rivest, Shamir, Adleman, 1976]

For slightly over 3 weeks, each day Rivest and Shamir came up with a new scheme to do public-key cryptography, . . . , and **by the next morning** Adleman had broken it.
The 23rd scheme, **Adleman couldn't break**.

This is the **RSA Public-Key Algorithm** that is used today in the **SSL algorithm**

Lets your browser **generate key** to send order to Amazon

without, **we believe**, divulging any **useful** information about your credit card number, or what you bought.

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$E_B(x) \equiv x^{\mathbf{e}} \pmod{pq}$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$E_B(x) \equiv x^{\mathbf{e}} \pmod{pq}$$

$$D_B(x) \equiv x^{\mathbf{d}} \pmod{pq}$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$\begin{aligned} E_B(x) &\equiv x^{\mathbf{e}} \pmod{pq} \\ D_B(x) &\equiv x^{\mathbf{d}} \pmod{pq} \\ D_B(E_B(m)) &\equiv (m^e)^d \pmod{pq} \end{aligned}$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$\begin{aligned}
E_B(x) &\equiv x^{\mathbf{e}} \pmod{pq} \\
D_B(x) &\equiv x^{\mathbf{d}} \pmod{pq} \\
D_B(E_B(m)) &\equiv (m^e)^d \pmod{pq} \\
&\equiv m^{1-k\varphi(pq)} \pmod{pq}
\end{aligned}$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$\begin{aligned}
E_B(x) &\equiv x^{\mathbf{e}} && (\bmod\ pq) \\
D_B(x) &\equiv x^{\mathbf{d}} && (\bmod\ pq) \\
D_B(E_B(m)) &\equiv (m^e)^d && (\bmod\ pq) \\
&\equiv m^{1-k\varphi(pq)} && (\bmod\ pq) \\
&\equiv m \cdot (m^{\varphi(pq)})^{-k} && (\bmod\ pq)
\end{aligned}$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$\begin{aligned}
E_B(x) &\equiv x^{\mathbf{e}} \pmod{pq} \\
D_B(x) &\equiv x^{\mathbf{d}} \pmod{pq} \\
D_B(E_B(m)) &\equiv (m^e)^d \pmod{pq} \\
&\equiv m^{1-k\varphi(pq)} \pmod{pq} \\
&\equiv m \cdot (m^{\varphi(pq)})^{-k} \pmod{pq} \\
&\equiv m \pmod{pq} \quad \textbf{by Euler's Thm}
\end{aligned}$$

RSA

B **chooses** p, q n-bit primes, and e, s.t. $\gcd(e, \varphi(pq)) = 1$

B **publishes**: pq, e; keeps p, q secret.

Using Euclid's algorithm, B **computes** d, k, s.t.

$$ed + k\varphi(pq) = 1 \qquad [\varphi(pq) = (p-1)(q-1)].$$

[Break message into pieces shorter than $2n$ bits]

$$\begin{aligned}
E_B(x) &\equiv x^{\mathbf{e}} &&(\bmod\ pq) \\
D_B(x) &\equiv x^{\mathbf{d}} &&(\bmod\ pq) \\
D_B(E_B(m)) &\equiv (m^e)^d &&(\bmod\ pq) \\
&\equiv m^{1-k\varphi(pq)} &&(\bmod\ pq) \\
&\equiv m \cdot (m^{\varphi(pq)})^{-k} &&(\bmod\ pq) \\
&\equiv m &&(\bmod\ pq) \quad \textbf{by Euler's Thm} \\
&\equiv E_B(D_B(m)) &&(\bmod\ pq)
\end{aligned}$$

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],
It is **widely believed**: $E_B(m)$ divulges **no useful information** about m to anyone not knowing p, q, or d.

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],
It is **widely believed**: $E_B(m)$ divulges **no useful information** about m to anyone not knowing p, q, or d.

Message signing:

Let $m =$ "B promises to give A \$10, valid until 12/17/16."

Let $m' = m, r$ where r is nonce or current date and time.

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],
It is **widely believed**: $E_B(m)$ divulges **no useful information** about m to anyone not knowing p, q, or d.

Message signing:

Let $m =$ "B promises to give A \$10, valid until 12/17/16."

Let $m' = m, r$ where r is nonce or current date and time.

It is **widely believed** $D_B(m')$ could be produced only by B.

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],
It is **widely believed**: $E_B(m)$ divulges **no useful information** about m to anyone not knowing p, q, or d.

Message signing:

Let $m =$ "B promises to give A \$10, valid until 12/17/16."

Let $m' = m, r$ where r is nonce or current date and time.

It is **widely believed** $D_B(m')$ could be produced only by B.

Thus it can be used as a **contract** signed by B.

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],
It is **widely believed**: $E_B(m)$ divulges **no useful information** about m to anyone not knowing p, q, or d.

Message signing:
Let $m =$ "B promises to give A \$10, valid until 12/17/16."

Let $m' = m, r$ where r is nonce or current date and time.

It is **widely believed** $D_B(m')$ could be produced only by B.

Thus it can be used as a **contract** signed by B.

Useful for proving **authenticity**.

For **sufficiently large** n, [$n \geq 1000$ bits is currently fine],
It is **widely believed**: $E_B(m)$ divulges **no useful information** about m to anyone not knowing p, q, or d.

Message signing:
Let $m =$ "B promises to give A \$10, valid until 12/17/16."

Let $m' = m, r$ where r is nonce or current date and time.

It is **widely believed** $D_B(m')$ could be produced only by B.

Thus it can be used as a **contract** signed by B.

Useful for proving **authenticity**.

Public Key Cryptography is a **theoretical underpinning** for possible computer security even over the web.