Finite fields, Permutation Polynomials. Computational aspects with applications to public key cryptography

King Fahd University of Petroleum and Minerals

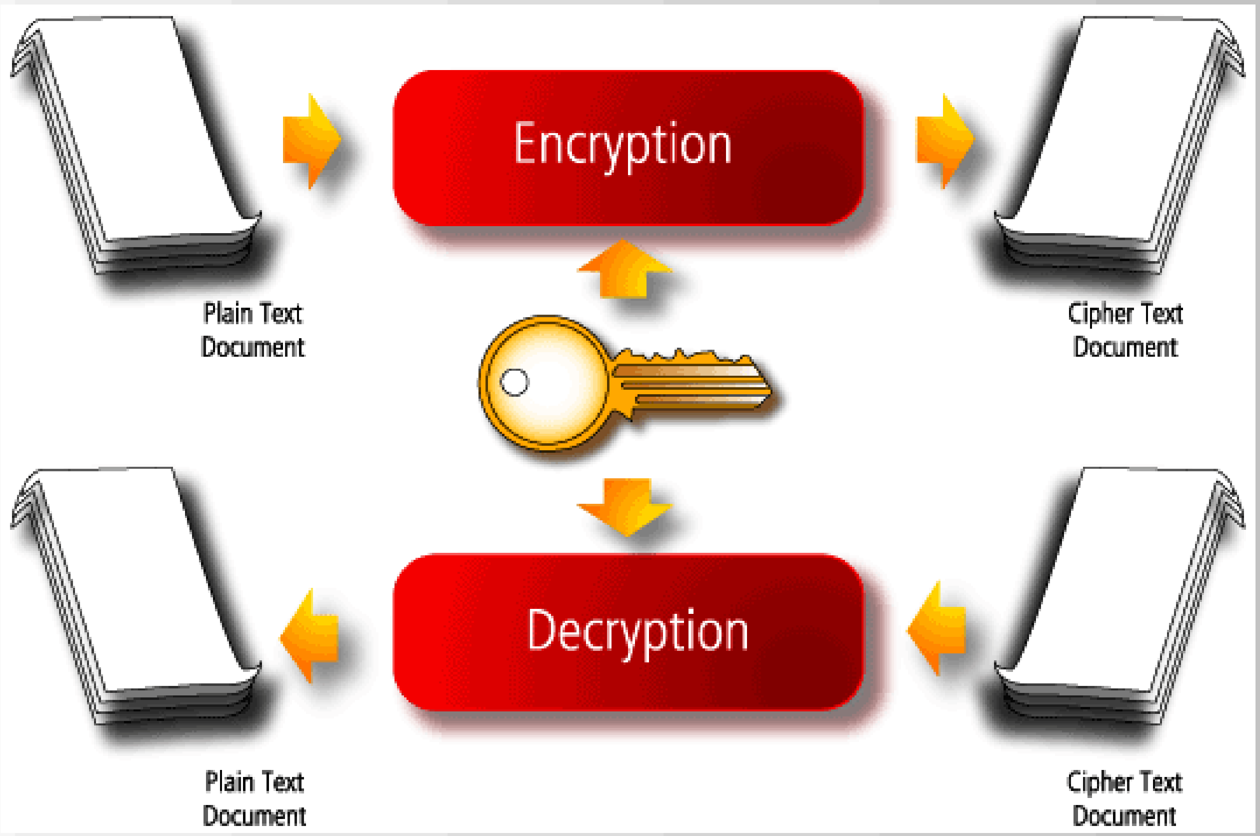
Dhahran, Saudi Arabia

Workshop on Industrial Mathematics

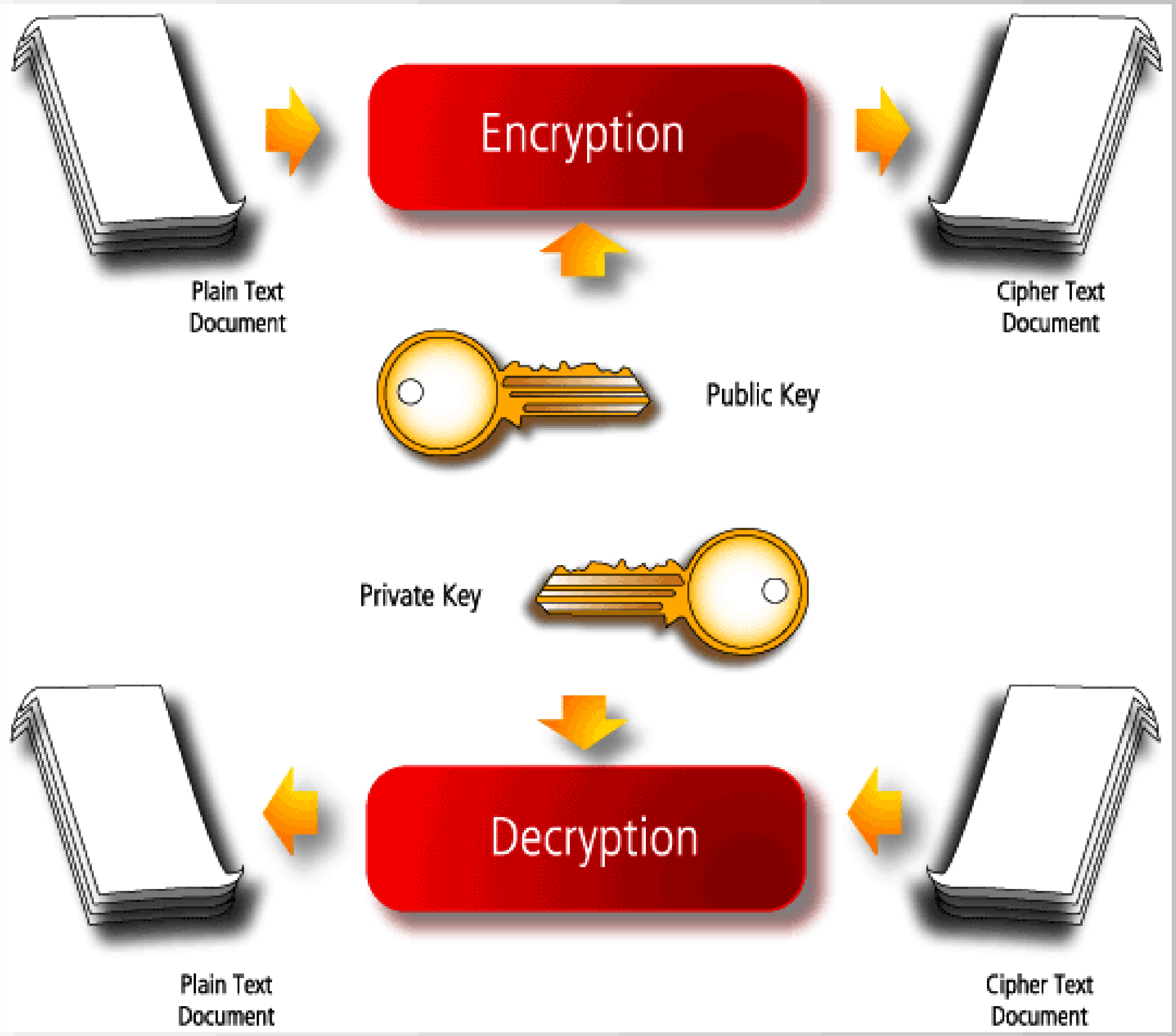
March 1, 2004

Private key versus Public Key

Private key versus Public Key



Private key versus Public Key



Classical General Examples of PKC

❶

❷

❸

Classical General Examples of PKC

1. (1976) Diffie Hellmann Key exchange protocol
2.
3.

Classical General Examples of PKC

1. (1976) Diffie Hellmann Key exchange protocol
   IEEE Trans. Information Theory IT-22 (1976)
2.
3.

Classical General Examples of PKC

1. (1976) Diffie Hellmann Key exchange protocol
   IEEE Trans. Information Theory IT-22 (1976)
2. (1983) Massey Omura Cryptosystem
3.

Classical General Examples of PKC

1. (1976) Diffie Hellmann Key exchange protocol
   IEEE Trans. Information Theory IT-22 (1976)
2. (1983) Massey Omura Cryptosystem
   Proc. 4^{th} Benelux Symposium on Information Theory (1983)
3.

Classical General Examples of PKC

1. (1976) Diffie Hellmann Key exchange protocol
   IEEE Trans. Information Theory IT-22 (1976)
2. (1983) Massey Omura Cryptosystem
   Proc. 4^{th} Benelux Symposium on Information Theory (1983)
3. (1984) ElGamal Cryptosystem

Classical General Examples of PKC

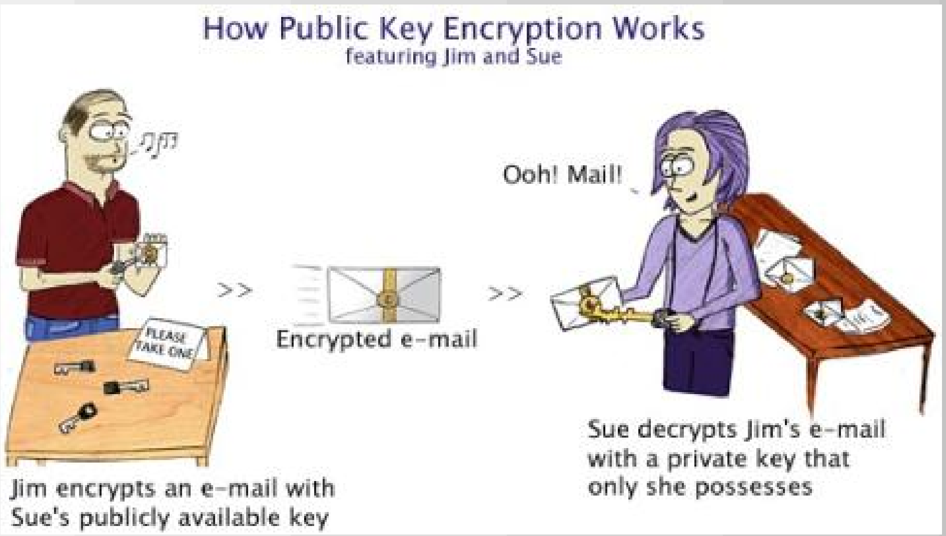
1. (1976) Diffie Hellmann Key exchange protocol
   IEEE Trans. Information Theory IT-22 (1976)
2. (1983) Massey Omura Cryptosystem
   Proc. 4^{th} Benelux Symposium on Information Theory (1983)
3. (1984) ElGamal Cryptosystem
   IEEE Trans. Information Theory IT-31 (1985)

Classical General Examples of PKC

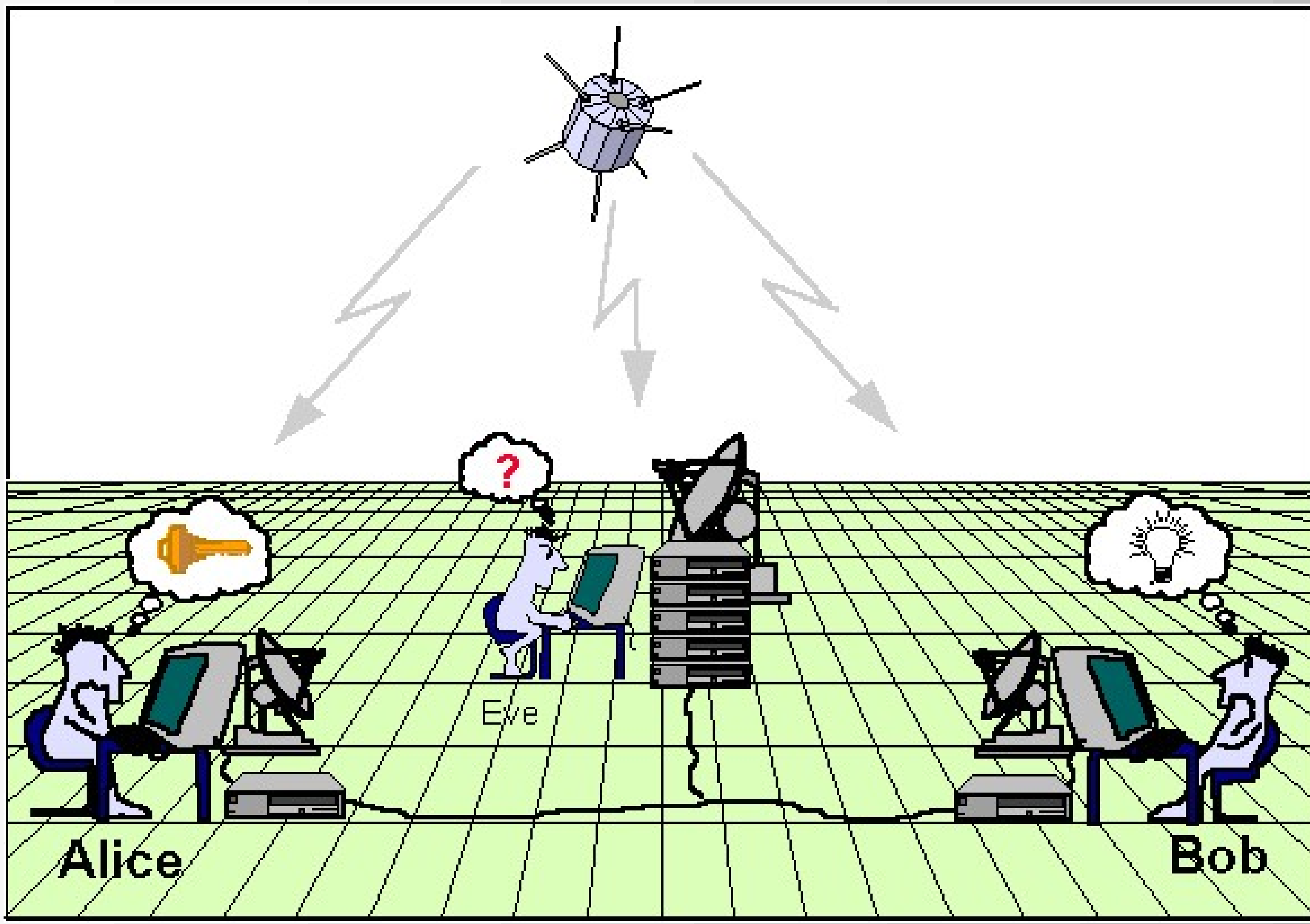
❶ (1976) Diffie Hellmann Key exchange protocol
IEEE Trans. Information Theory IT-22 (1976)

❷ (1983) Massey Omura Cryptosystem
Proc. 4^{th} Benelux Symposium on Information Theory (1983)

❸ (1984) ElGamal Cryptosystem
IEEE Trans. Information Theory IT-31 (1985)



Diffie–Hellmann key exchange 1/5



Diffie–Hellmann key exchange 2/5

Diffie–Hellmann key exchange 2/5

1.
2.
3.
4.
5.

Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2.
3.
4.
5.

Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a,
3.
4.
5.

Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a, $0 \leq a \leq p-1$
3.
4.
5.

Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a, $0 \le a \le p-1$
3. **Bob** picks a secret b, $0 \le b \le p-1$
4.
5.

Diffie–Hellmann key exchange 2/5

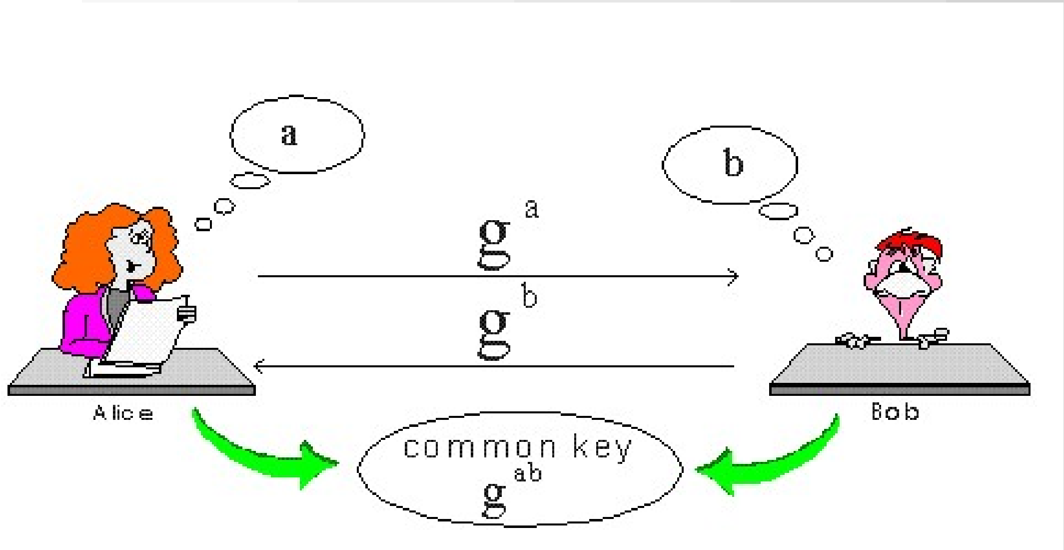
1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a, $0 \le a \le p-1$
3. **Bob** picks a secret b, $0 \le b \le p-1$
4. They compute and publish $g^a \bmod p$ (**Alice**) and $g^b \bmod p$ (**Bob**)
5.

Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a, $0 \leq a \leq p-1$
3. **Bob** picks a secret b, $0 \leq b \leq p-1$
4. They compute and publish $g^a \bmod p$ (**Alice**) and $g^b \bmod p$ (**Bob**)
5. The common secret key is $g^{ab} \bmod p$

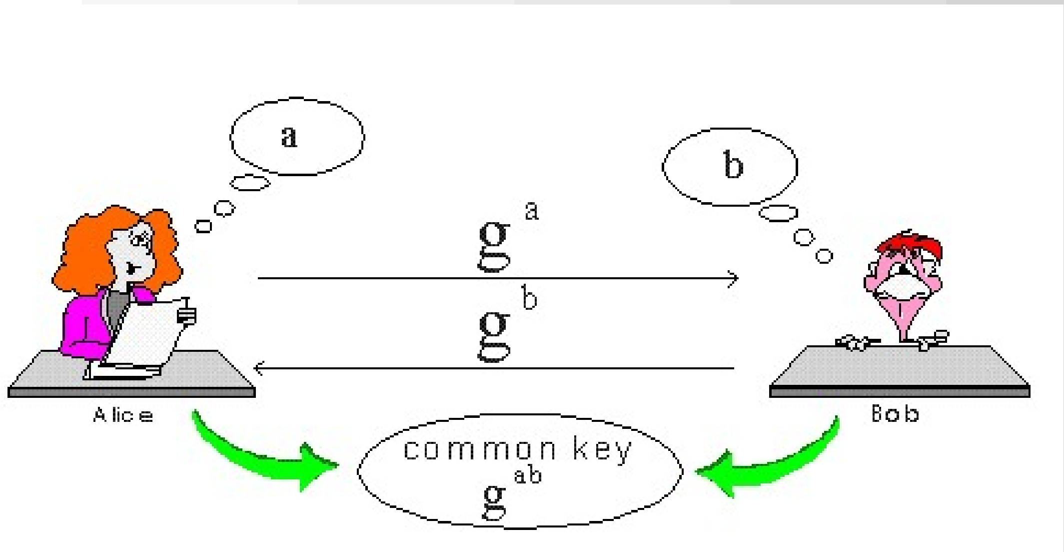
Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a, $0 \leq a \leq p-1$
3. **Bob** picks a secret b, $0 \leq b \leq p-1$
4. They compute and publish $g^a \bmod p$ (**Alice**) and $g^b \bmod p$ (**Bob**)
5. The common secret key is $g^{ab} \bmod p$



Diffie–Hellmann key exchange 2/5

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Alice** picks a secret a, $0 \leq a \leq p-1$
3. **Bob** picks a secret b, $0 \leq b \leq p-1$
4. They compute and publish $g^a \bmod p$ (**Alice**) and $g^b \bmod p$ (**Bob**)
5. The common secret key is $g^{ab} \bmod p$



what is a generator of $\mathbb{Z}/p\mathbb{Z}$?

Diffie–Hellmann key exchange 3/5

- ☞
- ☞
- ☞
- ☞
- ☞

Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number
-
-
-
-

Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
-
-
-
-

Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
- So $g \bmod p, \; g^2 \bmod p, \ldots, \; g^{p-1} \bmod p$ are all distinct,
-
-
-

Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
- So $g \bmod p,\ g^2 \bmod p, \ldots,\ g^{p-1} \bmod p$ are all distinct, i.e., a permutation of 1 through $p-1$
-
-
-

Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
- So $g \bmod p$, $g^2 \bmod p, \ldots,$ $g^{p-1} \bmod p$ are all distinct, i.e., a permutation of 1 through $p-1$
- In other words: for all $b \in \mathbb{Z}/p\mathbb{Z}, b \neq 0,$
-
-

Diffie–Hellman key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
- So $g \bmod p$, $g^2 \bmod p, \ldots,$ $g^{p-1} \bmod p$ are all distinct, i.e., a permutation of 1 through $p-1$
- In other words: for all $b \in \mathbb{Z}/p\mathbb{Z}, b \neq 0$, there exists an exponent $i \in \{0, 1, \ldots, p-1\}$ such that $b = g^i \bmod p$
-
-

Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \dots, p-1$
- So g mod p, g^2 mod $p, \dots,$ g^{p-1} mod p are all distinct, i.e., a permutation of 1 through $p-1$
- In other words: for all $b \in \mathbb{Z}/p\mathbb{Z}, b \neq 0$, there exists an exponent $i \in \{0, 1, \dots, p-1\}$ such that $b = g^i \bmod p$
- Given $b \in \mathbb{Z}$, exponent i above is
-

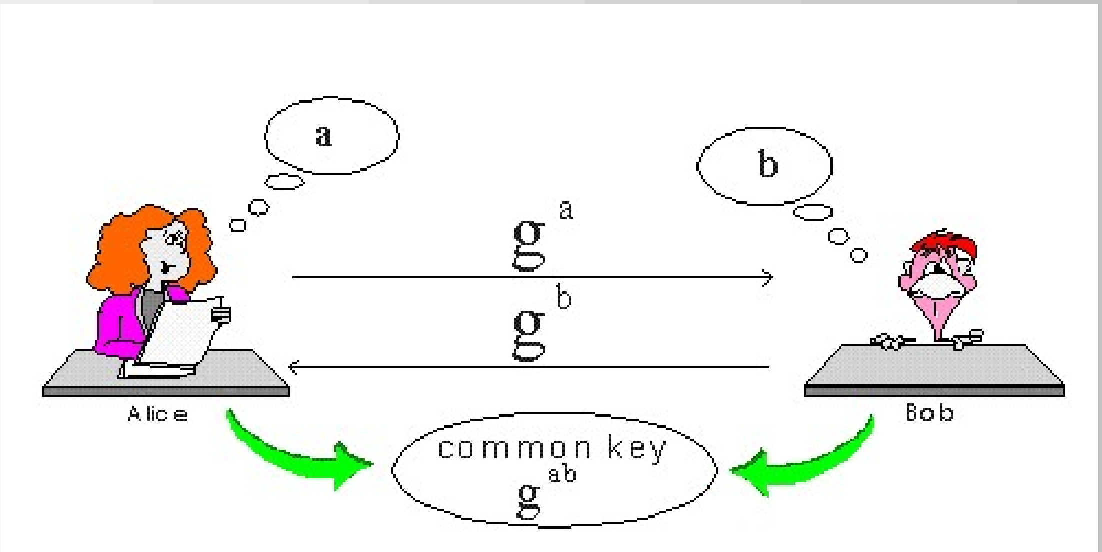
Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
- So g mod p, g^2 mod $p, \ldots,$ g^{p-1} mod p are all distinct, i.e., a permutation of 1 through $p-1$
- In other words: for all $b \in \mathbb{Z}/p\mathbb{Z}, b \neq 0$, there exists an exponent $i \in \{0, 1, \ldots, p-1\}$ such that $b = g^i \bmod p$
- Given $b \in \mathbb{Z}$, exponent i above is the discrete logarithm of b for base $g \bmod p$
-

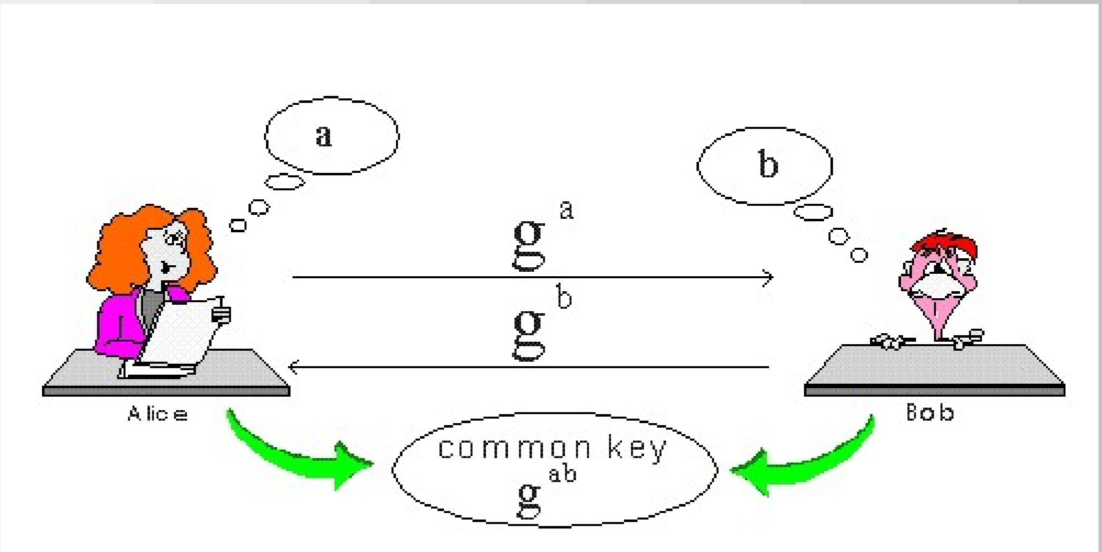
Diffie–Hellmann key exchange 3/5

- A generator (or primitive root) g of a prime number p is a number whose powers mod p, generate $1, \ldots, p-1$
- So g mod p, g^2 mod $p, \ldots,$ g^{p-1} mod p are all distinct, i.e., a permutation of 1 through $p-1$
- In other words: for all $b \in \mathbb{Z}/p\mathbb{Z}, b \neq 0$, there exists an exponent $i \in \{0, 1, \ldots, p-1\}$ such that $b = g^i \bmod p$
- Given $b \in \mathbb{Z}$, exponent i above is the discrete logarithm of b for base $g \bmod p$
- Computing discrete logs appears infeasible in general

Diffie–Hellmann key exchange 4/5



Diffie–Hellmann key exchange 4/5

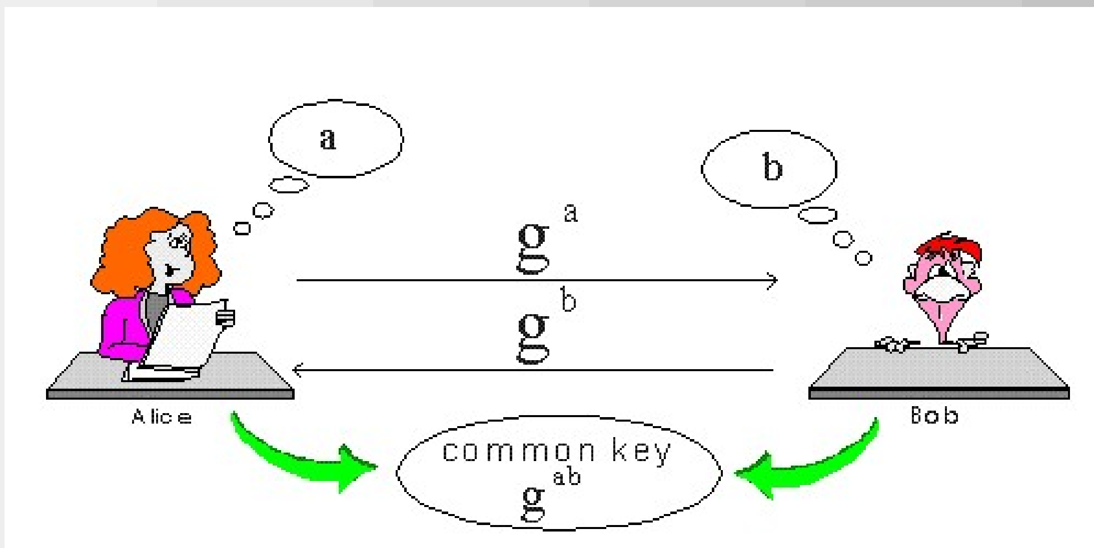


☞ ;

☞

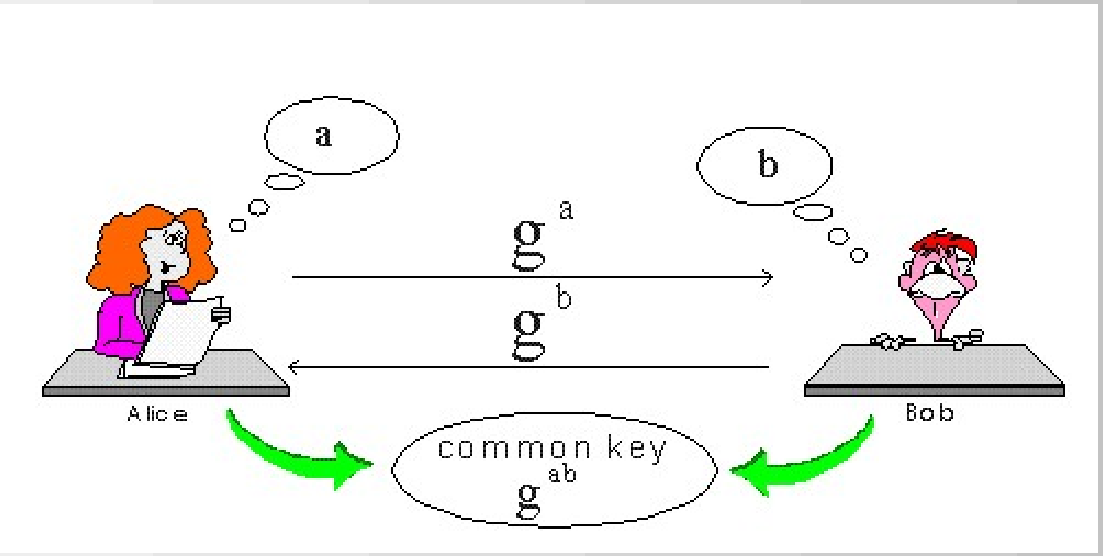
☞

Diffie–Hellmann key exchange 4/5



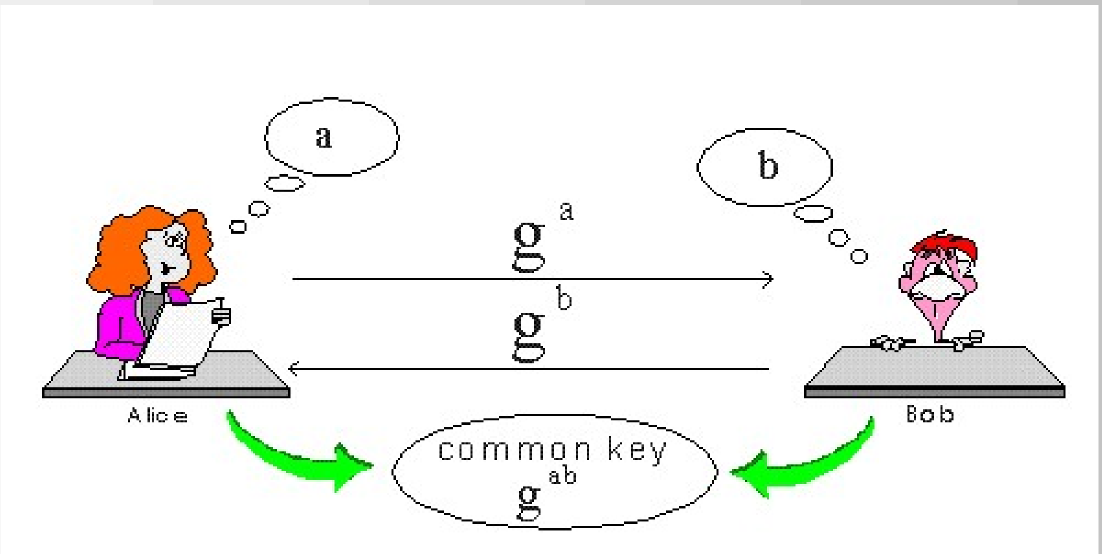
- **Eve** knows g^a, g^b but would like to compute g^{ab};
-
-

Diffie–Hellmann key exchange 4/5



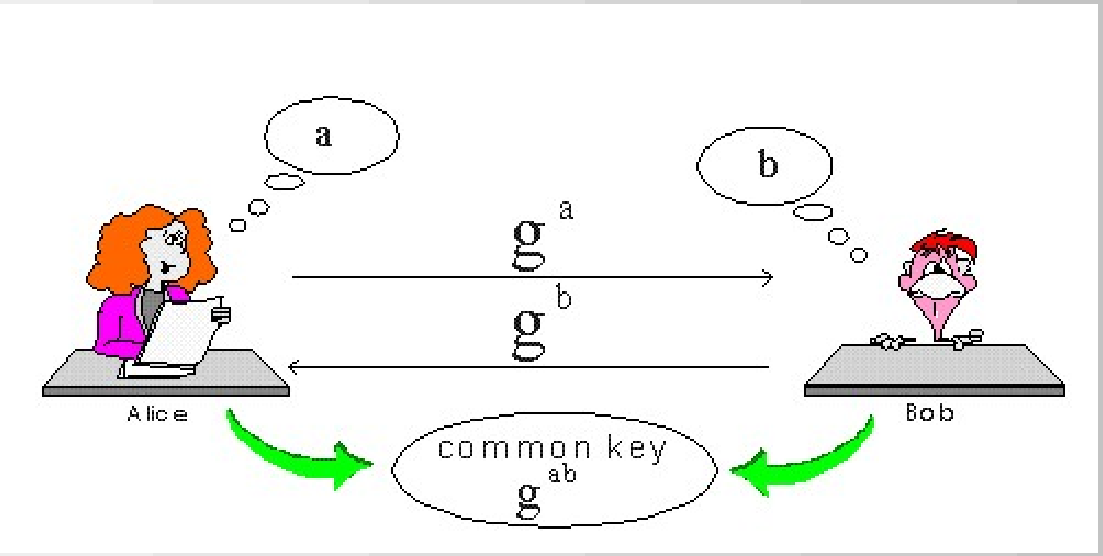
- **Eve** knows g^a, g^b but would like to compute g^{ab};
- **Eve** could compute a discrete logarithm to find a and then $(g^b)^a$
-

Diffie–Hellmann key exchange 4/5



- **Eve** knows g^a, g^b but would like to compute g^{ab};
- **Eve** could compute a discrete logarithm to find a and then $(g^b)^a$
- for given α, g, p, **Eve** should solve:

Diffie–Hellmann key exchange 4/5



- **Eve** knows g^a, g^b but would like to compute g^{ab};
- **Eve** could compute a discrete logarithm to find a and then $(g^b)^a$
- for given α, g, p, **Eve** should solve:

$$g^X \equiv \alpha \bmod p$$

Diffie–Hellmann key exchange 5/5

Diffie–Hellmann key exchange 5/5

A “criptographically meaningful size” example:

Diffie–Hellmann key exchange 5/5

A “criptographically meaningful size” example:

$p = 370273307460967425842481081357528298315386585184169353328410050632472746552261503118421027658$
$721711241508544733578984012456938357678209461867245573821426204444288523552318347549870943602$
$190239876925965853744436584289 0327$

Diffie–Hellmann key exchange 5/5

A “criptographically meaningful size” example:

$p = 370273307460967425842481081357528298315386585184169353328410050632472746552261503118421027658$
$7217112415085447335789840124569383576782094618672455738214262044442885235523183475498709436 02$
$1902398769259658537444365842890327$

$g = 5$

Diffie–Hellmann key exchange 5/5

A "criptographically meaningful size" example:

$p =$ 3702733074609674258424810813575282983153865851841693533284100506324727465522615031184210276587217112415085447335789840124569383576782094618672455738214262044442885235523183475498709436021902398769259658537444365842890327

$g = 5$

$a =$ 2308840902039895388227917479653026722679565668038909847198111704018348814235352410395561538395030079070601651217032418664096044274135079002294214909329210457060330466911747378679898500024210343154844771162635809902530822

Diffie–Hellmann key exchange 5/5

A “criptographically meaningful size” example:

p = 370273307460967425842481081357528298315386585184169353328410050632472746552261503118421027658
721711241508544733578984012456938357678209461867245573821426204444288523552318347549870943602
190239876925965853744436584289032

$g = 5$

a = 230884090203989538822791747965302672226795656680389098471981117040183488142353524103955615383\
9
503007907060165121703241866409604427413507900229421490932921045706033046691174737867989850
002421034315484477116263580990253082

b = 202628627712040976052737350793757540205242681192017941068774728007392912193775762330719406560
040933311164190467406050768556042798567906868136988403326100887782675574881508824219596637
05180574380470308541288799465419522 89

Diffie–Hellmann key exchange 5/5

A “criptographically meaningful size” example:

$p =$ 37027330746096742584248108135752829831538658518416935332841005063247274655226150311842102765872171124150854473357898401245693835767820946186724557382142620444428852355523183475498709436021902398769259658537444365842890327

$g = 5$

$a =$ 2308840902039895388227917479653026722679565668038909847198111704018348814235352410395561538395030079070601651217032418664096044274135079002294214909329210457060330466911747378679898500024210343154844771162635809902530822

$b =$ 2026286277120409760527373507937575402052426811920179410687747280073929121937757623307194065600409333111641904674060507685560427985679068681369884033261008877826755748815088242195966370518057438047030854128879946541952289

$5^a =$ 2494514241078932628924844425756891566223499407710247477336124609623103292094965304814697324109595757601247732395287229562052325375814376804042234303084056865342398577185839357814166518479146351026737882783508710913577680

Diffie–Hellmann key exchange 5/5

A "criptographically meaningful size" example:

$p =$ 370273307460967425842481081357528298315386585184169353328410050632472746552261503118421027658
721711241508544733578984012456938357678209461867245573821426204444288523552318347549870943602
19023987692596585374443658428903 27

$g = 5$

$a =$ 230884090203989538822791747965302672226795656680389098471981117040183488142353524103955615 3839
503007907060165121703241866409604427413507900229421490932921045706033046691174737867 98985
00024210343154844771162635809902530822

$b =$ 202628627712040976052737350793757540205242681192017941068774728007392912193775762330719406560
040933311164190467406050768556042798567906868136988403326100887782675574881508824219596 63
70518057438047030854128879946541952289

$5^a =$ 249451424107893262892484442575689156622349940771024747733612460962310329209496530481469732410
959575760124773239528722956205232537581437680404223430308405686534239857718583935781416 65
18479146351026737882783508710913577680

$5^b =$ 287293760357523957032946092556813694596882586743260552838382768832192594422702357607546631218
640014853957893014446177932232015947060973983603311951612138362147414988242010983310457 62
16804562648795943563091024975401008295

Diffie–Hellmann key exchange 5/5

A "criptographically meaningful size" example:

$p =$ 37027330746096742584248108135752829831538658518416935332841005063247274655226150311842102765872171124150854473357898401245693835767820946186724557382142620444428852355231834754987094360219023987692596585374443658428903 27

$g = 5$

$a =$ 23088409020398953882279174796530267226795656680389098471981117040183488142353524103955615383950300790706016512170324186640960442741350790022942149093292104570603304669117473786798985000242103431548447711626358099025308 22

$b =$ 20262862771204097605273735079375754020524268119201794106877472800739291219377576233071940656004093331116419046740605076855604279856790686813698840332610088778267557488150882421959663705180574380470308541288799465419522 89

$5^a =$ 24945142410789326289248444257568915662234994077102474773361246096231032920949653048146973241095957576012477323952872295620523253758143768040422343030840568653423985771858393578141665184791463510267378827835087109135776 80

$5^b =$ 28729376035752395703294609255681369459688258674326055283838276883219259442270235760754663121864001485395789301444617793223201594706097398360331195161213836214741498824201098331045762168045626487959435630910249754010082 95

$5^{ab} =$ 36674172125349300306071275329964633749875664216293811088694156172838197865927916343627669411439682348921744440103868565092597181273385376288526293344498755858906626836268436664512871223950829209587369115457329515844644 96

Discrete Logarithms Computation

Discrete Logarithms Computation

Some classical algorithms:

Discrete Logarithms Computation

Some classical algorithms:

-
-
-
-

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
-
-
-

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} *Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).*
-
-
-

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} *Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).*
- Pohlig–Hellmann Algorithm
-
-

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).
- Pohlig–Hellmann Algorithm
  IEEE Trans. Information Theory IT-24 (1978).
-
-

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} *Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).*
- Pohlig–Hellmann Algorithm
  IEEE Trans. Information Theory IT-24 (1978).
- Index computation algorithm
-

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} *Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).*
- Pohlig–Hellmann Algorithm
  IEEE Trans. Information Theory IT-24 (1978).
- Index computation algorithm
- Sieving algorithms

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} *Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).*
- Pohlig–Hellmann Algorithm
  IEEE Trans. Information Theory IT-24 (1978).
- Index computation algorithm
- Sieving algorithms
  La Macchia & Odlyzko, Designs Codes and Cryptography 1 (1991)

Discrete Logarithms Computation

Some classical algorithms:

- Shanks baby-step, giant step
  Proc. 2^{nd} Manitoba Conf. Numerical Mathematics (Winnipeg, 1972).
- Pohlig–Hellmann Algorithm
  IEEE Trans. Information Theory IT-24 (1978).
- Index computation algorithm
- Sieving algorithms
  La Macchia & Odlyzko, Designs Codes and Cryptography 1 (1991)

NOTE: The last two are "very special" for $\mathbb{Z}/p\mathbb{Z}$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$\begin{aligned} p &= \lfloor 10^{89}\pi \rfloor + 156137 \\ &= 314159265358979323846264338327950288419716939937510582097494459230781640628620899862959619, \\ & g = 2, \end{aligned}$$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$
\begin{aligned}
p &= \lfloor 10^{89}\pi \rfloor + 156137 \\
&= 314159265358979323846264338327950288419716939937510582097494459230781640628620899862959619, \\
& \quad g = 2, \\
y &= \lfloor 10^{89} e \rfloor \\
&= 271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217
\end{aligned}
$$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$
\begin{aligned}
p &= \lfloor 10^{89}\pi \rfloor + 156137 \\
&= 31415926535897932384626433832795028841971693993751058209749445923078164062862089986295961 9, \\
&\quad g = 2, \\
y &= \lfloor 10^{89} e \rfloor \\
&= 271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217
\end{aligned}
$$

$$2^X \equiv y \bmod p$$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$\begin{aligned} p &= \lfloor 10^{89}\pi \rfloor + 156137 \\ &= 31415926535897932384626433832795028841971693993751058209749445923078164062862089986295961 9, \\ & g = 2, \end{aligned}$$

$$\begin{aligned} y &= \lfloor 10^{89} e \rfloor \\ &= 27182818284590452353602874713526624977572470936999595749669676277240766303535475945713821 7 \end{aligned}$$

$$2^X \equiv y \bmod p$$

$$y = g^{1767138072114216962732048234071620272302057952449914157493844716677918658538374188101093},$$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$\begin{aligned} p &= \lfloor 10^{89}\pi \rfloor + 156137 \\ &= 31415926535897932384626433832795028841971693993751058209749445923078164062862089986295961 9, \\ & g = 2, \end{aligned}$$

$$\begin{aligned} y &= \lfloor 10^{89} e \rfloor \\ &= 2718281828459045235360287471352662497757247093699959574966967627724076630353547594571382 17 \end{aligned}$$

$$2^X \equiv y \bmod p$$

$$y = g^{1767138072114216962732048234071620272302057952449914157493844716677918658538374188101093},$$

$$y + 1 = g^{3116041987058269748820788091978682382044912000142161761705846865427122180292692723003342 1},$$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$\begin{aligned} p &= \lfloor 10^{89}\pi \rfloor + 156137 \\ &= 31415926535897932384626433832795028841971693993751058209749445923078164062862089986295961 9, \\ & \quad g = 2, \end{aligned}$$

$$\begin{aligned} y &= \lfloor 10^{89} e \rfloor \\ &= 27182818284590452353602874713526624977572470936999595749669676277240766303535475945713821 7 \end{aligned}$$

$$2^X \equiv y \bmod p$$

$$y = g^{1767138072114216962732048234071620272302057952449914157493844716677918658538374188101093},$$

$$y + 1 = g^{3116041987058269748820788091978682382044912000142161761705846865427122180292692723003342 1},$$

$$y + 2 = g^{3089883293350445253338277649145014072371680345775342279270337839998667742527392786788373 01},$$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$
\begin{aligned}
p &= \lfloor 10^{89}\pi \rfloor + 156137 \\
&= 314159265358979323846264338327950288419716939937510582097494459230781640628620899862959619, \\
& g = 2, \\
y &= \lfloor 10^{89} e \rfloor \\
&= 271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217
\end{aligned}
$$

$$2^X \equiv y \bmod p$$

$y = g^{1767138072114216962732048234071620272302057952449914157493844716677918658538374188101093}$,

$y+1 = g^{3116041987058269748820788091978682382044912000142161761705846865427122180292692723003342 1}$,

$y+2 = g^{3089883293350445253338277649145014072371680345775342279270337839998667742527392786788373 01}$,

$y+3 = g^{6580688800278838010371298688366325318718350540545118893505511320988794936425513481529784 6}$,

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$
\begin{aligned}
p &= \lfloor 10^{89}\pi \rfloor + 156137 \\
&= 31415926535897932384626433832795028841971693993751058209749445923078164062862089986295961 9, \\
& g = 2, \\
y &= \lfloor 10^{89} e \rfloor \\
&= 27182818284590452353602874713526624977572470936999595749669676277240766303535475945713821 7
\end{aligned}
$$

$$2^X \equiv y \bmod p$$

$y = g^{1767138072114216962732048234071620272302057952449914157493844716677918658538374188101093}$,

$y + 1 = g^{3116041987058269748820788091978682382044912000142161761705846865427122180292692723003342 1}$,

$y + 2 = g^{3089883293350445253338277649145014072371680345775342279270337839998667742527392786788373 01}$,

$y + 3 = g^{6580688800278838010371298688366325318718350540545118893505511320988794936425513481529784 6}$,

$y + 4 = g^{4069601088212869919975316593460491889486849045436061788784458793535379546218510507897709 3}$

Discrete Logarithms computation Records 1/2

A. Joux et R. Lercier, 1998.

$$\begin{aligned} p &= \lfloor 10^{89}\pi \rfloor + 156137 \\ &= 314159265358979323846264338327950288419716939937510582097494459230781640628620899862959619, \\ & \quad g = 2, \end{aligned}$$

$$\begin{aligned} y &= \lfloor 10^{89} e \rfloor \\ &= 271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217 \end{aligned}$$

$$2^X \equiv y \bmod p$$

$$y = g^{1767138072114216962732048234071620272302057952449914157493844716677918658538374188101093},$$

$$y + 1 = g^{3116041987058269748820788091978682382044912000142161761705846865427122180292692723003342 1},$$

$$y + 2 = g^{3089883293350445253338277649145014072371680345775342279270337839998667742527392786788373 01},$$

$$y + 3 = g^{6580688800278838010371298688366325318718350540545118893505511320988794936425513481529784 6},$$

$$y + 4 = g^{4069601088212869919975316593460491889486849045436061788784458793535379546218510507897709 3}$$

It took 4.5 months... on a Pentium PRO 180 MHz

Discrete Logarithms computation Records 2/2

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

①

②

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

②

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server`

②

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

②

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400`

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③ 2001 $p \cong 10^{120}$;

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$
`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$
`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③ 2001 $p \cong 10^{120}$; **(Current Record!)**

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③ 2001 $p \cong 10^{120}$; **(Current Record!)** `2.5 months`

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③ 2001 $p \cong 10^{120}$; **(Current Record!)** `2.5 months`

$p = \lfloor 10^{119}\pi \rfloor + 207819,\ g = 2$

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③ 2001 $p \cong 10^{120}$; **(Current Record!)** `2.5 months`

$p = \lfloor 10^{119}\pi \rfloor + 207819,\ g = 2$

$y = \lfloor 10^{119} \rfloor$

Discrete Logarithms computation Records 2/2

A. Joux et R. Lercier (CNRS / Ecole Polytechnique)

① 1999 $p \cong 10^{100}$

`500MHz quadri-processors Dec Alpha Server` – 8.5 months;

② 2001 $p \cong 10^{110}$

`525MHz quadri-processors Digital Alpha Server 8400` – 20 days;

③ 2001 $p \cong 10^{120}$; **(Current Record!)** `2.5 months`

$p = \lfloor 10^{119}\pi \rfloor + 207819,\ g = 2$

$y = \lfloor 10^{119} \rfloor$

$$y = g^{\substack{262112280685811387636008622038191827370390768520656974243035 \\ 380382193478767436018681449804940840373741641452864730765082}},$$

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

1.

2.

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2.

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Bob** picks a secret b, $0 < b \leq p - 1$,

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Bob** picks a secret b, $0 < b \leq p - 1$,
   he computes $\beta = g^b \bmod p$ and publishes β

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

1. **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$
2. **Bob** picks a secret b, $0 < b \leq p - 1$,
   he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

❶ **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$

❷ **Bob** picks a secret b, $0 < b \leq p - 1$,
he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

①

②

③

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

❶ **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$

❷ **Bob** picks a secret b, $0 < b \leq p-1$,
he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

① **Alice** picks a secret k,

②

③

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

❶ **Alice** and **Bob** agree on a prime p and a <u>*generator*</u> g in $\mathbb{Z}/p\mathbb{Z}$

❷ **Bob** picks a secret b, $0 < b \leq p - 1$,
he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

① **Alice** picks a secret k, $0 < k \leq p - 1$

②

③

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

❶ **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$

❷ **Bob** picks a secret b, $0 < b \leq p-1$,
he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

① **Alice** picks a secret k, $0 < k \leq p-1$

② She computes $\alpha = g^k \bmod p$ and $\gamma = x \cdot \beta^a \bmod p$

③

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

❶ **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$

❷ **Bob** picks a secret b, $0 < b \leq p - 1$,
he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

① **Alice** picks a secret k, $0 < k \leq p - 1$

② She computes $\alpha = g^k \bmod p$ and $\gamma = x \cdot \beta^a \bmod p$

③ The encrypted message is

ElGamal Cryptosystem 1/2

Alice wants to sent a message $x \in \mathbb{Z}/p\mathbb{Z}$ to **Bob**

SETUP:

❶ **Alice** and **Bob** agree on a prime p and a *generator* g in $\mathbb{Z}/p\mathbb{Z}$

❷ **Bob** picks a secret b, $0 < b \leq p - 1$,
he computes $\beta = g^b \bmod p$ and publishes β

ENCRYPTION: (Alice)

① **Alice** picks a secret k, $0 < k \leq p - 1$

② She computes $\alpha = g^k \bmod p$ and $\gamma = x \cdot \beta^a \bmod p$

③ The encrypted message is

$$E(x) = (\alpha, \gamma) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$$

ElGamal Cryptosystem 2/2

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

①

②

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

②

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

②

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

② It works because

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

② It works because

$$D(E(x)) = D(\alpha, \gamma) = x \cdot g^{bk} \cdot g^{k(p-1-b)} = x$$

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

② It works because

$$D(E(x)) = D(\alpha, \gamma) = x \cdot g^{bk} \cdot g^{k(p-1-b)} = x$$

since $g^{k(p-1)} \bmod p = 1$ by

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

② It works because

$$D(E(x)) = D(\alpha, \gamma) = x \cdot g^{bk} \cdot g^{k(p-1-b)} = x$$

since $g^{k(p-1)} \bmod p = 1$ by

Fermat Little Theorem If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

② It works because

$$D(E(x)) = D(\alpha, \gamma) = x \cdot g^{bk} \cdot g^{k(p-1-b)} = x$$

since $g^{k(p-1)} \bmod p = 1$ by

Fermat Little Theorem If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

Eve can decrypt the message if he can compute the discrete logarithm X,

ElGamal Cryptosystem 2/2

DECRYPTION: (Bob)

① **Bob** computes

$$D(\alpha, \gamma) = \gamma \cdot \alpha^{p-1-b} \bmod p$$

② It works because

$$D(E(x)) = D(\alpha, \gamma) = x \cdot g^{bk} \cdot g^{k(p-1-b)} = x$$

since $g^{k(p-1)} \bmod p = 1$ by

Fermat Little Theorem If p is prime, $p \nmid a \in \mathbb{N}$

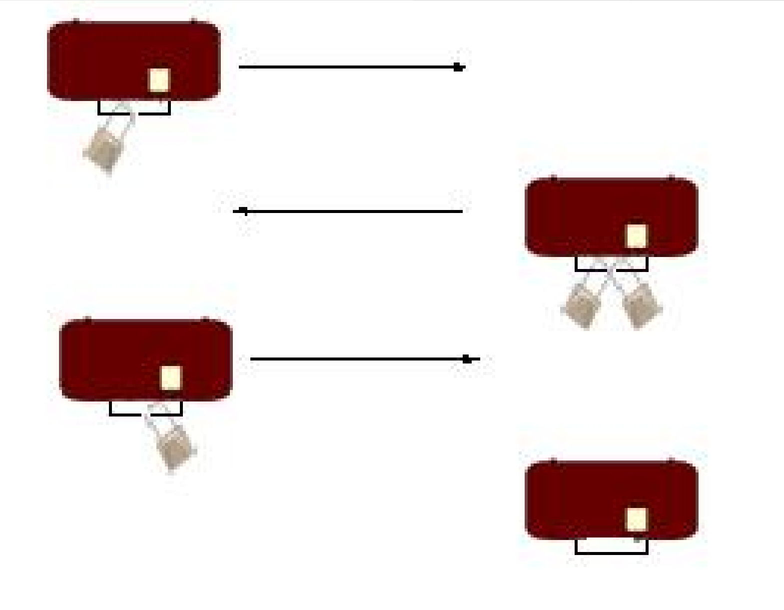
$$a^{p-1} \equiv 1 \bmod p$$

Eve can decrypt the message if he can compute the discrete logarithm X,

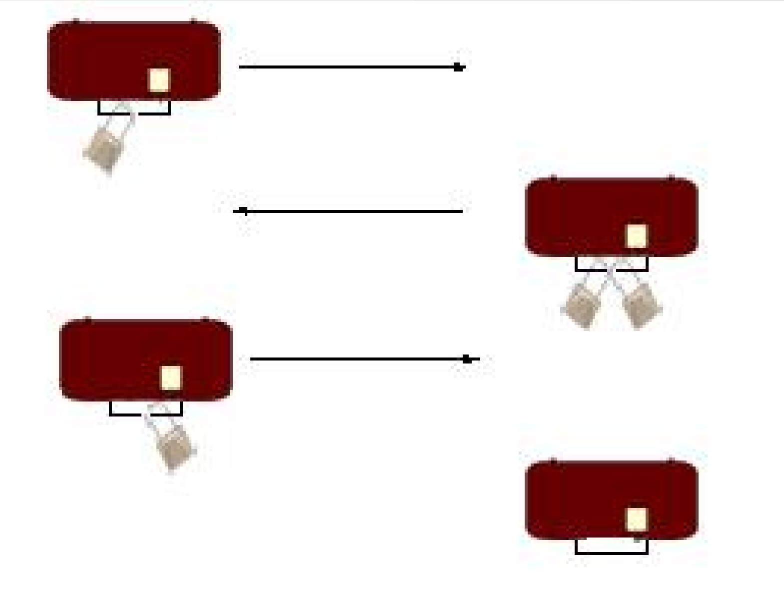
$$\beta = g^X \bmod p$$

Massey Omura 1/2

Massey Omura 1/2

Alice  Bob

Massey Omura 1/2

Alice  **Bob**

①

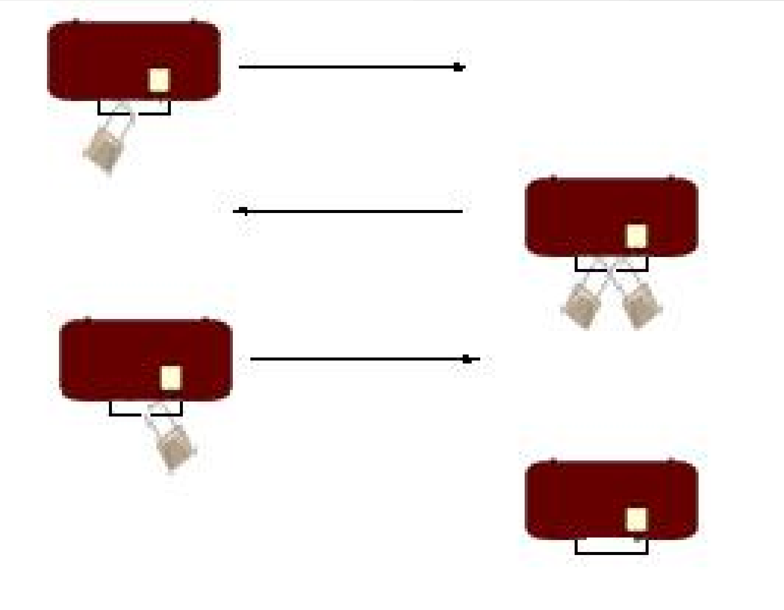
②

③

④

⑤

Massey Omura 1/2

Alice  **Bob**

① **Alice** and **Bob** each picks a secret key $k_A, k_B \in \{1, \ldots, p-1\}$

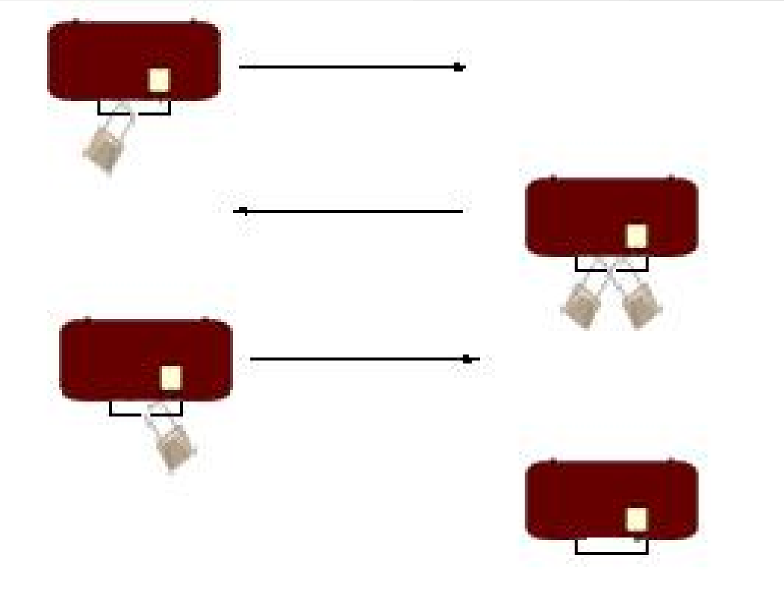
②

③

④

⑤

Massey Omura 1/2

Alice  **Bob**

① **Alice** and **Bob** each picks a secret key $k_A, k_B \in \{1, \ldots, p-1\}$

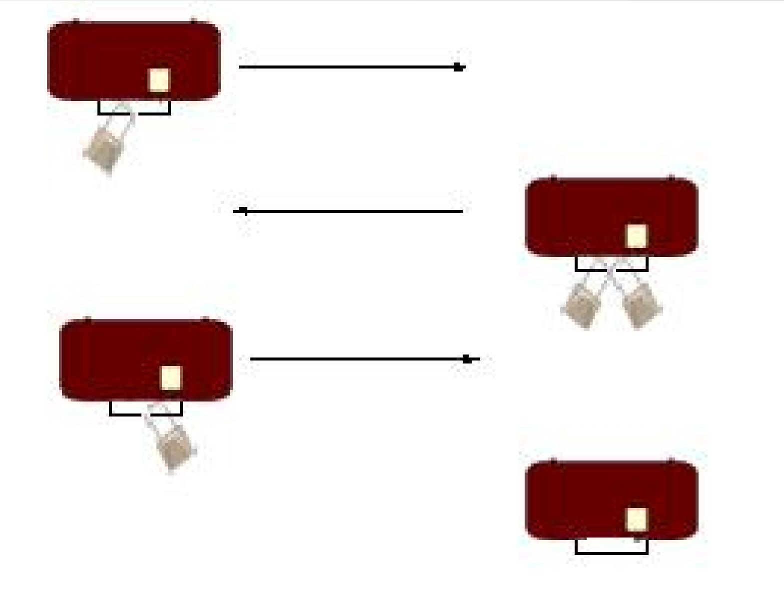
② They compute $l_A, l_B \in \{1, \ldots, p-1\}$ such that

③

④

⑤

Massey Omura 1/2

Alice  **Bob**

① **Alice** and **Bob** each picks a secret key $k_A, k_B \in \{1, \dots, p-1\}$

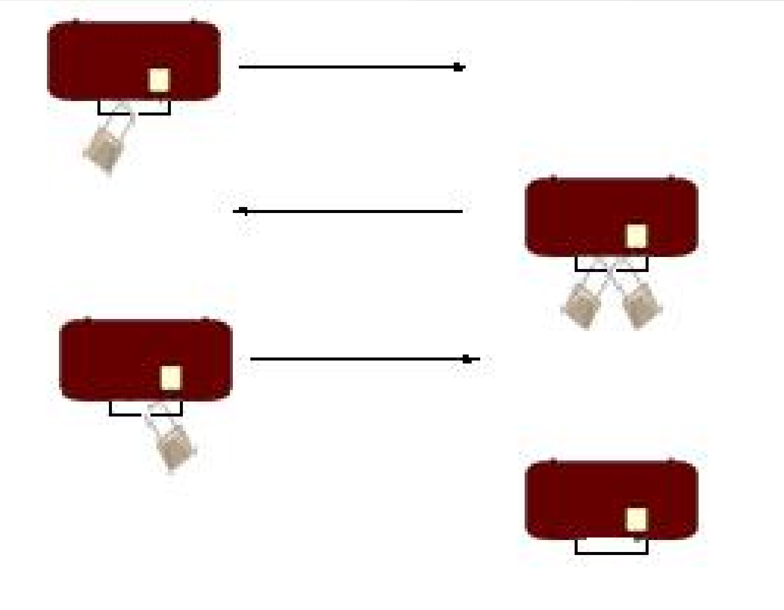
② They compute $l_A, l_B \in \{1, \dots, p-1\}$ such that

③ $k_A l_A = 1(\text{mod} p-1)$ and $k_B l_B = 1(\text{mod} p-1)$

④

⑤

Massey Omura 1/2

Alice  **Bob**

① **Alice** and **Bob** each picks a secret key $k_A, k_B \in \{1, \dots, p-1\}$

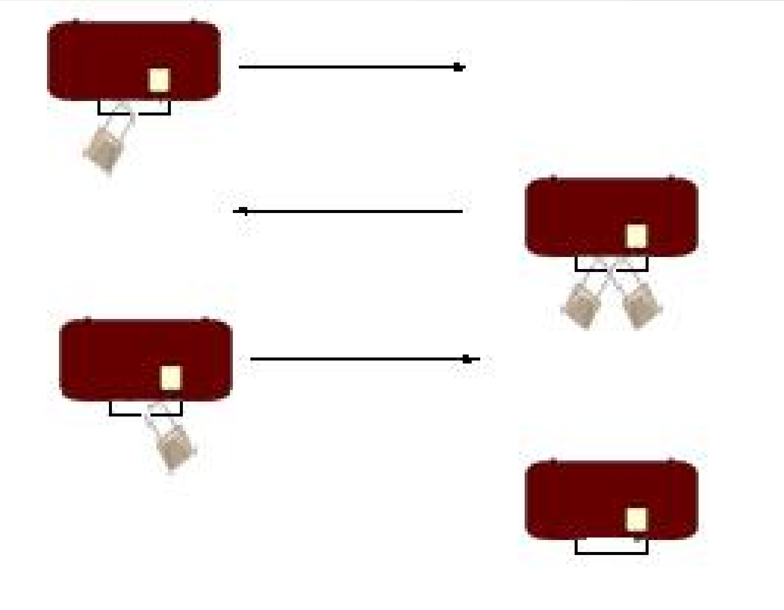
② They compute $l_A, l_B \in \{1, \dots, p-1\}$ such that

③ $k_A l_A = 1(\bmod p-1)$ and $k_B l_B = 1(\bmod p-1)$

④ **Alice** key is (k_A, l_A) (k_A to lock and l_A to unlock)

⑤

Massey Omura 1/2



Alice Bob

① **Alice** and **Bob** each picks a secret key $k_A, k_B \in \{1, \ldots, p-1\}$

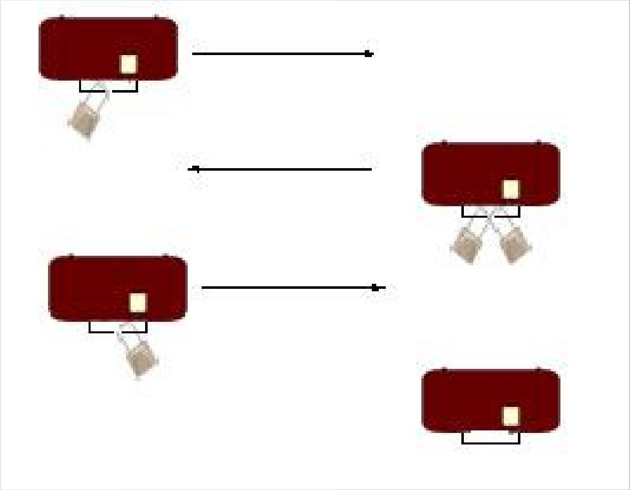
② They compute $l_A, l_B \in \{1, \ldots, p-1\}$ such that

③ $k_A l_A = 1(\text{mod} p - 1)$ and $k_B l_B = 1(\text{mod} p - 1)$

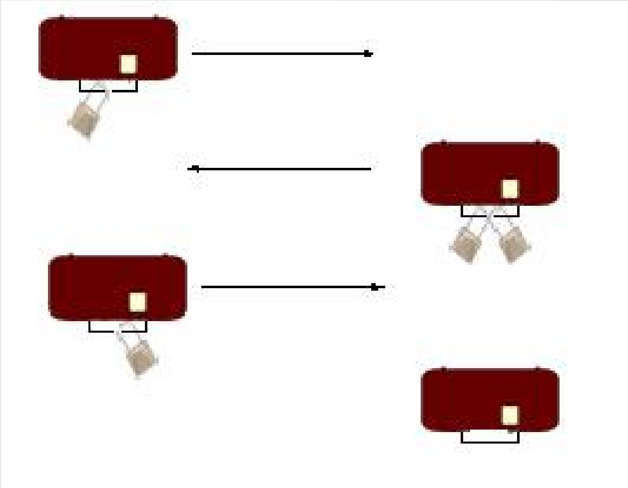
④ **Alice** key is (k_A, l_A) (k_A to lock and l_A to unlock)

⑤ **Bob** key is (k_B, l_B) (k_B to lock and l_B to unlock)

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

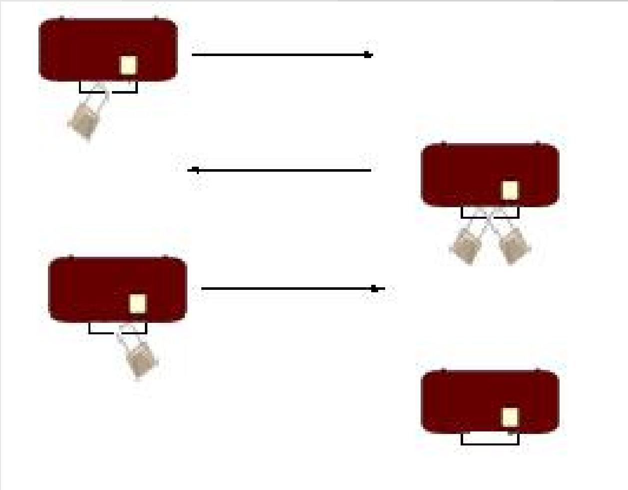
①

②

③

④

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

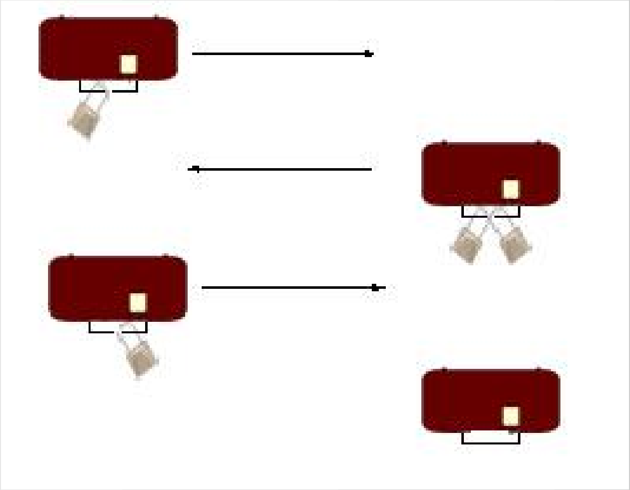
① To send the message P, **Alice** computes and sends $M = P^{k_A} \bmod p$

②

③

④

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

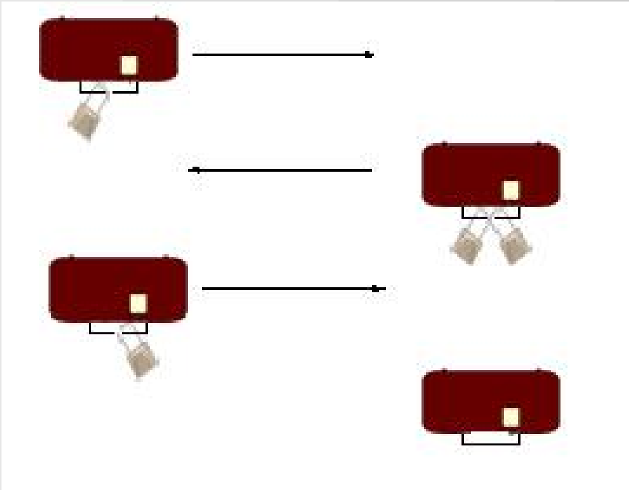
① To send the message P, **Alice** computes and sends $M = P^{k_A} \bmod p$

② **Bob** computes and sends back $N = M^{k_B} \bmod p$

③

④

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

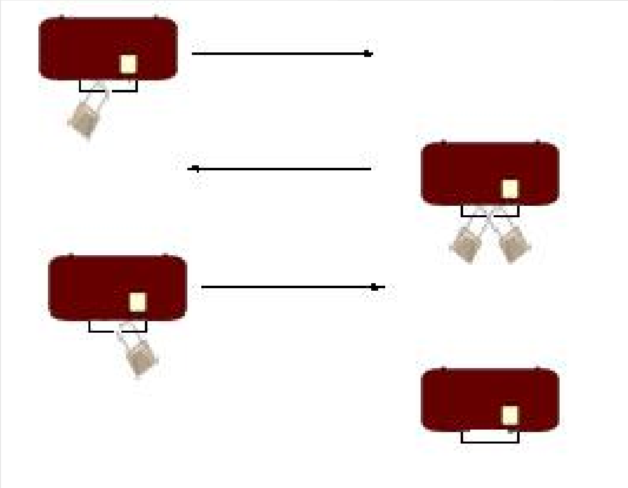
① To send the message P, **Alice** computes and sends $M = P^{k_A} \bmod p$

② **Bob** computes and sends back $N = M^{k_B} \bmod p$

③ **Alice** computes $L = N^{l_A} \pmod p$ and sends it back to **Bob**

④

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

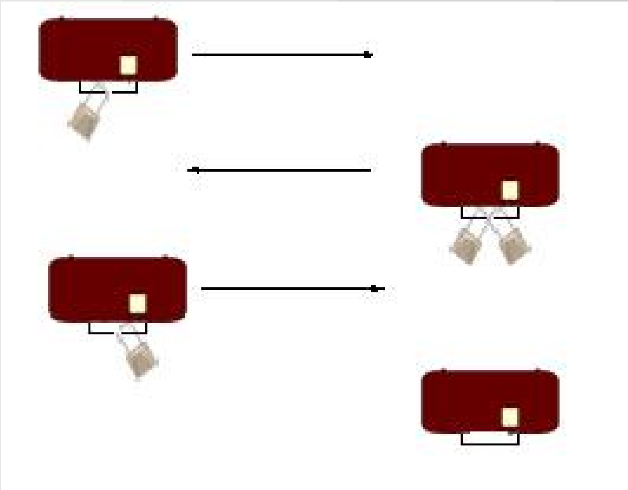
① To send the message P, **Alice** computes and sends $M = P^{k_A} \bmod p$

② **Bob** computes and sends back $N = M^{k_B} \bmod p$

③ **Alice** computes $L = N^{l_A} \pmod{p}$ and sends it back to **Bob**

④ **Bob** decrypt the message computing $P = L^{l_B} \pmod{p}$

Massey Omura 2/2

Alice (k_A, l_A)  **Bob** (k_B, l_B)

① To send the message P, **Alice** computes and sends $M = P^{k_A} \bmod p$

② **Bob** computes and sends back $N = M^{k_B} \bmod p$

③ **Alice** computes $L = N^{l_A} \pmod p$ and sends it back to **Bob**

④ **Bob** decrypt the message computing $P = L^{l_B} \pmod p$

It works: $P = L^{l_B} = N^{l_A l_B} = M^{k_B l_A l_B} = P^{k_A k_B l_A l_B}$ by Fermat Little Theorem

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

Examples of cyclic groups

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

Examples of cyclic groups

①

②

③

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

Examples of cyclic groups

① Elliptic curves modulo p

②

③

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

Examples of cyclic groups

① Elliptic curves modulo p

② Multiplicative groups of Finite Fields

③

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

Examples of cyclic groups

① Elliptic curves modulo p

② Multiplicative groups of Finite Fields

③ Dickson Polynomials over finite fields

From $\mathbb{Z}/p\mathbb{Z}$ to cyclic groups

We can substitute $\mathbb{Z}/p\mathbb{Z}$ with a set G where it is possible to compute powers P^a and there is a generator (there is $g \in G$ such that for each $\alpha \in G$, $\alpha = g^i$ for a suitable i); cyclic groups

Examples of cyclic groups

① Elliptic curves modulo p

② Multiplicative groups of Finite Fields

③ Dickson Polynomials over finite fields

Finite Fields

Finite Fields

☞

☞

☞

☞

☞

☞

☞

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
-
-
-
-
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$
-
-
-
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

-
-
-
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

- $\mathbb{F}_p[x]/(f)$ is a field
-
-
-
-

Finite Fields

- ☞ Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- ☞ Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

- ☞ $\mathbb{F}_p[x]/(f)$ is a field

  $(g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f)$

- ☞
- ☞
- ☞
- ☞

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}$$

- $\mathbb{F}_p[x]/(f)$ is a field

  ($g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f$)
- $\mathbb{F}_p[x]/(f)$ does not depend on f
-
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible ($m = \partial(f)$)

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

- $\mathbb{F}_p[x]/(f)$ is a field

  ($g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f$)
- $\mathbb{F}_p[x]/(f)$ does not depend on f

  (i.e. if $h \in \mathbb{F}_p[x]$ irreducible, $\partial f = \partial h \implies \mathbb{F}_p[x]/(f) \cong \mathbb{F}_p[x]/(h)$)
-
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

- $\mathbb{F}_p[x]/(f)$ is a field

  $(g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f)$
- $\mathbb{F}_p[x]/(f)$ does not depend on f

  (i.e. if $h \in \mathbb{F}_p[x]$ irreducible, $\partial f = \partial h \implies \mathbb{F}_p[x]/(f) \cong \mathbb{F}_p[x]/(h)$)
- $$\boxed{\mathbb{F}_{p^m} = \mathbb{F}_p[x]/(f)}$$
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}$$

- $\mathbb{F}_p[x]/(f)$ is a field
  ($g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f$)
- $\mathbb{F}_p[x]/(f)$ does not depend on f
  (i.e. if $h \in \mathbb{F}_p[x]$ irreducible, $\partial f = \partial h \implies \mathbb{F}_p[x]/(f) \cong \mathbb{F}_p[x]/(h)$)
- $$\mathbb{F}_{p^m} = \mathbb{F}_p[x]/(f)$$
  any choice of f with $m = \partial f$ is the same
-
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

- $\mathbb{F}_p[x]/(f)$ is a field
  $(g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f)$
- $\mathbb{F}_p[x]/(f)$ does not depend on f
  (i.e. if $h \in \mathbb{F}_p[x]$ irreducible, $\partial f = \partial h \implies \mathbb{F}_p[x]/(f) \cong \mathbb{F}_p[x]/(h)$)
- $$\boxed{\mathbb{F}_{p^m} = \mathbb{F}_p[x]/(f)}$$
  any choice of f with $m = \partial f$ is the same
- $|\mathbb{F}_{p^m}| = p^m$
-

Finite Fields

- Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\}$ (field if p prime)
- Given $f \in \mathbb{F}_p[x]$ irreducible $(m = \partial(f))$

$$\boxed{\mathbb{F}_p[x]/(f) = \{a_0 + a_1 t + \cdots + a_{m-1} t^{m-1} \mid a_i \in \mathbb{F}_p\}}$$

- $\mathbb{F}_p[x]/(f)$ is a field

  ($g_1 \star g_2 \in \mathbb{F}_p[x]/(f)$ is $g_1 g_2 \bmod f$)
- $\mathbb{F}_p[x]/(f)$ does not depend on f

  (i.e. if $h \in \mathbb{F}_p[x]$ irreducible, $\partial f = \partial h \implies \mathbb{F}_p[x]/(f) \cong \mathbb{F}_p[x]/(h)$)
- $$\boxed{\mathbb{F}_{p^m} = \mathbb{F}_p[x]/(f)}$$

  any choice of f with $m = \partial f$ is the same
- $|\mathbb{F}_{p^m}| = p^m$
- $\mathbb{F}^*_{p^m} = \mathbb{F}_{p^m} \setminus \{0\}$ is a cyclic group under multiplication

Producing $\mathbb{F}_q$

Producing $\mathbb{F}_q$

Set $q = p^m$

Producing $\mathbb{F}_q$

Set $q = p^m$

- ☞
- ☞
- ☞
- ☞
- ☞

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \iff$ find $f \in I_m(q)$
-
-
-
-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \iff$ find $f \in I_m(q)$

  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$
-
-
-
-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \Longleftrightarrow$ find $f \in I_m(q)$

  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$

- $\sum_{d|m} d|I_d(q)| = q^m$

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \iff$ find $f \in I_m(q)$
  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$
- $\sum_{d|m} d|I_d(q)| = q^m$
- $|I_m(q)| = \frac{q^m - q}{m}$ (if m is prime) $|I_m(q)| \sim \frac{q^m}{m}$
-
-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \iff$ find $f \in I_m(q)$
  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$
- $\sum_{d|m} d|I_d(q)| = q^m$
- $|I_m(q)| = \frac{q^m - q}{m}$ (if m is prime) $|I_m(q)| \sim \frac{q^m}{m}$
- Some fields of cryptographic size:
-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \Longleftrightarrow$ find $f \in I_m(q)$

  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$

- $\sum_{d|m} d|I_d(q)| = q^m$

- $|I_m(q)| = \frac{q^m - q}{m}$ (if m is prime) $|I_m(q)| \sim \frac{q^m}{m}$

- Some fields of cryptographic size:

  $\mathbb{F}_{2^{503}} = \mathbb{F}_2[x]/(x^{503} + x^3 + 1), \mathbb{F}_{5323^{20}} = \mathbb{F}_{5323}[x]/(f)$

-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \iff$ find $f \in I_m(q)$

  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$

- $\sum_{d|m} d|I_d(q)| = q^m$

- $|I_m(q)| = \frac{q^m - q}{m}$ (if m is prime) $|I_m(q)| \sim \frac{q^m}{m}$

- Some fields of cryptographic size:

  $\mathbb{F}_{2^{503}} = \mathbb{F}_2[x]/(x^{503} + x^3 + 1), \mathbb{F}_{5323^{20}} = \mathbb{F}_{5323}[x]/(f)$

  $f = x^{20} + 1451x^{18} + 5202x^{17} + 752x^{16} + 3778x^{15} + 4598x^{14} + 2563x^{13} + 5275x^{12} + 4260x^{11} +$

-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \Longleftrightarrow$ find $f \in I_m(q)$
  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$
- $\sum_{d|m} d|I_d(q)| = q^m$
- $|I_m(q)| = \frac{q^m - q}{m}$ (if m is prime) $|I_m(q)| \sim \frac{q^m}{m}$
- Some fields of cryptographic size:
  $\mathbb{F}_{2^{503}} = \mathbb{F}_2[x]/(x^{503} + x^3 + 1), \mathbb{F}_{5323^{20}} = \mathbb{F}_{5323}[x]/(f)$
  $f = x^{20} + 1451x^{18} + 5202x^{17} + 752x^{16} + 3778x^{15} + 4598x^{14} + 2563x^{13} + 5275x^{12} + 4260x^{11} +$
  $862x^{10} + 4659x^9 + 3484x^8 + 1510x^7 + 4556x^6 + 2317x^5 + 2171x^4 + 3100x^3 + 4100x^2 + 682x + 5110$
-

Producing $\mathbb{F}_q$

Set $q = p^m$

- Produce $\mathbb{F}_q \iff$ find $f \in I_m(q)$

  $(I_m(q) = \{f \in \mathbb{F}_p[x], f \text{ irreducible}, \partial f = m\})$

- $\sum_{d|m} d|I_d(q)| = q^m$

- $|I_m(q)| = \frac{q^m - q}{m}$ (if m is prime) $|I_m(q)| \sim \frac{q^m}{m}$

- Some fields of cryptographic size:

  $\mathbb{F}_{2^{503}} = \mathbb{F}_2[x]/(x^{503} + x^3 + 1), \mathbb{F}_{5323^{20}} = \mathbb{F}_{5323}[x]/(f)$

  $f = x^{20} + 1451x^{18} + 5202x^{17} + 752x^{16} + 3778x^{15} + 4598x^{14} + 2563x^{13} + 5275x^{12} + 4260x^{11} +$

  $862x^{10} + 4659x^9 + 3484x^8 + 1510x^7 + 4556x^6 + 2317x^5 + 2171x^4 + 3100x^3 + 4100x^2 + 682x + 5110$

- Good to find f sparse

Interpolation on $\mathbb{F}_q$

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \longrightarrow \mathbb{F}_q$ a function.

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \to \mathbb{F}_q$ a function.

h can always be interpolated with a polynomial in $\mathbb{F}_q[x]$!

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \to \mathbb{F}_q$ a function.

h can always be interpolated with a polynomial in $\mathbb{F}_q[x]$!

☞ LAGRANGE INTERPOLATION

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \rightarrow \mathbb{F}_q$ a function.

h can always be interpolated with a polynomial in $\mathbb{F}_q[x]$!

☞ LAGRANGE INTERPOLATION

$$f_h(x) = \sum_{c \in \mathbb{F}_q} h(c) \prod_{\substack{d \in \mathbb{F}_q \\ d \neq c}} \frac{x - d}{c - d} \in \mathbb{F}_q[x]$$

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \to \mathbb{F}_q$ a function.

h can always be interpolated with a polynomial in $\mathbb{F}_q[x]$!

☞ Lagrange interpolation

$$f_h(x) = \sum_{c \in \mathbb{F}_q} h(c) \prod_{\substack{d \in \mathbb{F}_q \\ d \neq c}} \frac{x - d}{c - d} \in \mathbb{F}_q[x]$$

☞ Finite fields interpolation

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \to \mathbb{F}_q$ a function.

h can always be interpolated with a polynomial in $\mathbb{F}_q[x]$!

☞ LAGRANGE INTERPOLATION

$$f_h(x) = \sum_{c \in \mathbb{F}_q} h(c) \prod_{\substack{d \in \mathbb{F}_q \\ d \neq c}} \frac{x-d}{c-d} \in \mathbb{F}_q[x]$$

☞ FINITE FIELDS INTERPOLATION

$$f_h(x) = \sum_{c \in \mathbb{F}_q} h(c) \left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x]$$

Interpolation on $\mathbb{F}_q$

Given $h : \mathbb{F}_q \to \mathbb{F}_q$ a function.

h can always be interpolated with a polynomial in $\mathbb{F}_q[x]$!

☞ LAGRANGE INTERPOLATION

$$f_h(x) = \sum_{c \in \mathbb{F}_q} h(c) \prod_{\substack{d \in \mathbb{F}_q \\ d \neq c}} \frac{x-d}{c-d} \in \mathbb{F}_q[x]$$

☞ FINITE FIELDS INTERPOLATION

$$f_h(x) = \sum_{c \in \mathbb{F}_q} h(c) \left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x]$$

$$d^{q-1} = \begin{cases} 1 & d \neq 0 \\ 0 & d = 0 \end{cases}$$

More on interpolation in $\mathbb{F}_q$

More on interpolation in $\mathbb{F}_q$

☞ If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

More on interpolation in $\mathbb{F}_q$

☞ If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\boxed{\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)}$$

More on interpolation in $\mathbb{F}_q$

- If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\boxed{\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)}$$

- The interpolant polynomial is unique mod $x^q - x$

More on interpolation in $\mathbb{F}_q$

- If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\boxed{\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)}$$

- The interpolant polynomial is unique mod $x^q - x$

$$\boxed{\Longrightarrow \quad \text{unique with degree} \leq q-1}$$

More on interpolation in $\mathbb{F}_q$

- If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\boxed{\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)}$$

- The interpolant polynomial is unique mod $x^q - x$

$$\boxed{\Longrightarrow \quad \text{unique with degree} \leq q - 1}$$

- If $c_h = \#\{c \in \mathbb{F}_q \mid h(c) \neq c\}$,

More on interpolation in $\mathbb{F}_q$

- If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)$$

- The interpolant polynomial is unique mod $x^q - x$

$$\Longrightarrow \quad \text{unique with degree} \leq q - 1$$

- If $c_h = \#\{c \in \mathbb{F}_q \mid h(c) \neq c\}$,

$$q - c_h \leq \partial f_h \leq q - 2$$

More on interpolation in $\mathbb{F}_q$

- If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\boxed{\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)}$$

- The interpolant polynomial is unique mod $x^q - x$

$$\boxed{\Longrightarrow \quad \text{unique with degree} \leq q - 1}$$

- If $c_h = \#\{c \in \mathbb{F}_q \mid h(c) \neq c\}$,

$$\boxed{q - c_h \leq \partial f_h \leq q - 2}$$

- **Problem.** Find functions with sparse interpolation polynomial

More on interpolation in $\mathbb{F}_q$

- If $f_1, f_2 \in \mathbb{F}_q[x]$ with $f_1(c) = f_2(c) \forall c \in \mathbb{F}_q$,

$$\Longrightarrow \quad x^q - x \mid f_1(x) - f_2(x)$$

- The interpolant polynomial is unique mod $x^q - x$

$$\Longrightarrow \quad \text{unique with degree} \leq q - 1$$

- If $c_h = \#\{c \in \mathbb{F}_q \mid h(c) \neq c\}$,

$$q - c_h \leq \partial f_h \leq q - 2$$

- **Problem.** Find functions with sparse interpolation polynomial

Better if they are $\rightsquigarrow$Permutation polynomials

Permutation polynomials

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \longrightarrow \mathbb{F}_q \mid \sigma(1:1)\}$$

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

☞ $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \rightarrow \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

☞ $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

"f (as a funtion) is a permutation"

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \rightarrow \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

☞ $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

"f (as a funtion) is a permutation"

(i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

- $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if
  "f (as a funtion) is a permutation"
  (i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)
- If $f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c)\left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x] \Longrightarrow$

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

- ☞ $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

  "f (as a funtion) is a permutation"

  (i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)

- ☞ If $f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c)\left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x] \Longrightarrow$

  $$\boxed{f \in \mathbb{F}_q[x] \text{ is PP} \Longleftrightarrow \quad \exists \sigma \in \mathcal{S}(\mathbb{F}_q), f \equiv f_\sigma \bmod x^q - x}$$

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

- $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

  "f (as a funtion) is a permutation"

  (i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)

- If $f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c)\left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x] \Longrightarrow$

  $$\boxed{f \in \mathbb{F}_q[x] \text{ is PP} \iff \exists \sigma \in \mathcal{S}(\mathbb{F}_q), f \equiv f_\sigma \bmod x^q - x}$$

- **Examples:**

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

- $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

  "f (as a funtion) is a permutation"

  (i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)

- If $f_\sigma(x) = \sum_{c\in\mathbb{F}_q} \sigma(c)\left(1-(x-c)^{q-1}\right) \in \mathbb{F}_q[x] \Longrightarrow$

  $$\boxed{f \in \mathbb{F}_q[x] \text{ is PP} \iff \exists \sigma \in \mathcal{S}(\mathbb{F}_q), f \equiv f_\sigma \bmod x^q - x}$$

- **Examples:**

✎

✎

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

☞ $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

"f (as a funtion) is a permutation"

(i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)

☞ If $f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c) \left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x] \Longrightarrow$

$$\boxed{f \in \mathbb{F}_q[x] \text{ is PP} \iff \exists \sigma \in \mathcal{S}(\mathbb{F}_q), f \equiv f_\sigma \bmod x^q - x}$$

☞ **Examples:**

✎ $ax + b, \qquad a, b \in \mathbb{F}_q, a \neq 0$

✎

Permutation polynomials

$$\mathcal{S}(\mathbb{F}_q) = \{\sigma : \mathbb{F}_q \to \mathbb{F}_q \mid \sigma(1:1)\}$$

permutations of $\mathbb{F}_q$

- $f \in \mathbb{F}_q[x]$ is called permutation polynomial (PP) if

  "f (as a funtion) is a permutation"

  (i.e. $\exists \sigma \in \mathcal{S}(\mathbb{F}_q), \sigma(c) = f(c)\ \forall c \in \mathbb{F}_q$)

- If $f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c)\left(1 - (x-c)^{q-1}\right) \in \mathbb{F}_q[x] \Longrightarrow$

  $$\boxed{f \in \mathbb{F}_q[x] \text{ is PP} \iff \exists \sigma \in \mathcal{S}(\mathbb{F}_q), f \equiv f_\sigma \bmod x^q - x}$$

- **Examples:**
  - $ax + b, \qquad a, b \in \mathbb{F}_q, a \neq 0$
  - $x^k, \qquad (k, q-1) = 1$

More examples of PP

More examples of PP

✎ COMPOSITION. $f \circ g$ is PP if f, g are PP

More examples of PP

- ✎ **Composition.** $f \circ g$ is PP if f, g are PP
- ✎ $x^{(q+m-1)/m} + ax$ is a PP if $m|q-1$

More examples of PP

- ✎ COMPOSITION. $f \circ g$ is PP if f, g are PP
- ✎ $x^{(q+m-1)/m} + ax$ is a PP if $m|q-1$
- ✎ LINEARIZED POLYNOMIALS Let $q = p^m$,

More examples of PP

- ✎ COMPOSITION. $f \circ g$ is PP if f, g are PP
- ✎ $x^{(q+m-1)/m} + ax$ is a PP if $m | q-1$
- ✎ LINEARIZED POLYNOMIALS Let $q = p^m$,

$$L(x) = \sum_{s=0}^{r-1} \alpha_s x^{q^s} \qquad (\alpha_s \in \mathbb{F}_{p^m})$$

More examples of PP

- ✎ COMPOSITION. $f \circ g$ is PP if f, g are PP
- ✎ $x^{(q+m-1)/m} + ax$ is a PP if $m|q-1$
- ✎ LINEARIZED POLYNOMIALS Let $q = p^m$,

$$L(x) = \sum_{s=0}^{r-1} \alpha_s x^{q^s} \qquad (\alpha_s \in \mathbb{F}_{p^m})$$

➳

➳

More examples of PP

- COMPOSITION. $f \circ g$ is PP if f, g are PP
- $x^{(q+m-1)/m} + ax$ is a PP if $m | q-1$
- LINEARIZED POLYNOMIALS Let $q = p^m$,

$$L(x) = \sum_{s=0}^{r-1} \alpha_s x^{q^s} \qquad (\alpha_s \in \mathbb{F}_{p^m})$$

  - $L(c_1 + c_2) = L(c_1) + L(c_2)$
  -

More examples of PP

- ✎ COMPOSITION. $f \circ g$ is PP if f, g are PP
- ✎ $x^{(q+m-1)/m} + ax$ is a PP if $m|q-1$
- ✎ LINEARIZED POLYNOMIALS Let $q = p^m$,

$$\boxed{L(x) = \sum_{s=0}^{r-1} \alpha_s x^{q^s}} \qquad (\alpha_s \in \mathbb{F}_{p^m})$$

  - $L(c_1 + c_2) = L(c_1) + L(c_2)$
  - $L \in \mathrm{GL}_m(\mathbb{F}_p) \subset \mathcal{S}(\mathbb{F}_{p^m}) \iff \det(\alpha_{i-j}^{q^j}) \neq 0$

More examples of PP

- ✎ COMPOSITION. $f \circ g$ is PP if f, g are PP
- ✎ $x^{(q+m-1)/m} + ax$ is a PP if $m|q-1$
- ✎ LINEARIZED POLYNOMIALS Let $q = p^m$,

$$\boxed{L(x) = \sum_{s=0}^{r-1} \alpha_s x^{q^s}} \qquad (\alpha_s \in \mathbb{F}_{p^m})$$

  - ➽ $L(c_1 + c_2) = L(c_1) + L(c_2)$
  - ➽ $L \in \mathrm{GL}_m(\mathbb{F}_p) \subset \mathcal{S}(\mathbb{F}_{p^m}) \iff \det(\alpha_{i-j}^{q^j}) \neq 0$
    $\iff L(x) = 0$ has 1 solution

One more example of PP

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

$$D_k(x,a) = \sum_{j=0}^{[k/2]} \frac{k}{k-j} \binom{k-j}{j} (-a)^j x^{k-2j}$$

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

$$D_k(x,a) = \sum_{j=0}^{[k/2]} \frac{k}{k-j} \binom{k-j}{j} (-a)^j x^{k-2j}$$

➽

➽

➽

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

$$D_k(x,a) = \sum_{j=0}^{[k/2]} \frac{k}{k-j} \binom{k-j}{j} (-a)^j x^{k-2j}$$

- if $a \neq 0$, $D_k(x,a)$ is a PP $\iff$ $(k, q^2-1) = 1$
-
-

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

$$D_k(x,a) = \sum_{j=0}^{[k/2]} \frac{k}{k-j} \binom{k-j}{j} (-a)^j x^{k-2j}$$

- if $a \neq 0$, $D_k(x,a)$ is a PP $\iff (k, q^2-1) = 1$
- $D_k(x,0) = x^k$ is a PP $\iff (k, q-1) = 1$
-

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

$$D_k(x,a) = \sum_{j=0}^{[k/2]} \frac{k}{k-j} \binom{k-j}{j} (-a)^j x^{k-2j}$$

- if $a \neq 0$, $D_k(x,a)$ is a PP $\iff (k, q^2-1) = 1$
- $D_k(x,0) = x^k$ is a PP $\iff (k, q-1) = 1$
- **Note:** if $(mn, q^2-1) = 1$,

One more example of PP

✎ DICKSON POLYNOMIALS. If $a \in \mathbb{F}_q$, $k \in \mathbb{N}$

$$D_k(x,a) = \sum_{j=0}^{[k/2]} \frac{k}{k-j} \binom{k-j}{j} (-a)^j x^{k-2j}$$

- if $a \neq 0$, $D_k(x,a)$ is a PP $\iff$ $(k, q^2-1) = 1$
- $D_k(x,0) = x^k$ is a PP $\iff$ $(k, q-1) = 1$
- **Note:** if $(mn, q^2-1) = 1$,

$$D_m(D_n(x,\pm 1), \pm 1) = D_{mn}(x, \pm 1)$$

Dickson analogue of DH Key–exchange

Dickson analogue of DH Key–exchange

①

②

③

④

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

②

③

④

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

② **Alice** picks a secret $a \in [0, q^2 - 1]$, **Bob** picks a secret $b \in [0, q^2 - 1]$

③

④

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

② **Alice** picks a secret $a \in [0, q^2 - 1]$, **Bob** picks a secret $b \in [0, q^2 - 1]$

③ They compute and publish $D_a(\gamma, 1)$ (**Alice**) and $D_b(\gamma, 1)$ (**Bob**)

④

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

② **Alice** picks a secret $a \in [0, q^2 - 1]$, **Bob** picks a secret $b \in [0, q^2 - 1]$

③ They compute and publish $D_a(\gamma, 1)$ (**Alice**) and $D_b(\gamma, 1)$ (**Bob**)

④ The common secret key is

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

② **Alice** picks a secret $a \in [0, q^2 - 1]$, **Bob** picks a secret $b \in [0, q^2 - 1]$

③ They compute and publish $D_a(\gamma, 1)$ (**Alice**) and $D_b(\gamma, 1)$ (**Bob**)

④ The common secret key is

$$D_{ab}(\gamma, 1) = D_a(D_b(\gamma, 1, 1)) = D_b(D_a(\gamma, 1, 1))$$

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

② **Alice** picks a secret $a \in [0, q^2 - 1]$, **Bob** picks a secret $b \in [0, q^2 - 1]$

③ They compute and publish $D_a(\gamma, 1)$ (**Alice**) and $D_b(\gamma, 1)$ (**Bob**)

④ The common secret key is

$$D_{ab}(\gamma, 1) = D_a(D_b(\gamma, 1, 1)) = D_b(D_a(\gamma, 1, 1))$$

NOTE. There is a fast algorithm to compute the value of a Dickson polynomial at an element of $\mathbb{F}_q$

Dickson analogue of DH Key–exchange

① **Alice** and **Bob** agree on a finite field $\mathbb{F}_q$, and a generator $\gamma \in \mathbb{F}_q$

② **Alice** picks a secret $a \in [0, q^2 - 1]$, **Bob** picks a secret $b \in [0, q^2 - 1]$

③ They compute and publish $D_a(\gamma, 1)$ (**Alice**) and $D_b(\gamma, 1)$ (**Bob**)

④ The common secret key is

$$D_{ab}(\gamma, 1) = D_a(D_b(\gamma, 1, 1)) = D_b(D_a(\gamma, 1, 1))$$

NOTE. There is a fast algorithm to compute the value of a Dickson polynomial at an element of $\mathbb{F}_q$

Problem. Find new classes of PP

Enumeration of PP by degree

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- ☞
- ☞
- ☞
- ☞
- ☞

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- ☞ $\sum_{d \leq q-2} N_d(q) = q!$ $\qquad (\partial f_\sigma \leq q-2)$
- ☞
- ☞
- ☞
- ☞

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- $\sum_{d \leq q-2} N_d(q) = q!$ $\qquad (\partial f_\sigma \leq q-2)$
- $N_1(q) = q(q-1)$
-
-
-

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- $\sum_{d \leq q-2} N_d(q) = q!$ $(\partial f_\sigma \leq q-2)$
- $N_1(q) = q(q-1)$
- $N_d(q) = 0$ if $d | q-1$ (Hermite criterion)
-
-

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- $\sum_{d \leq q-2} N_d(q) = q!$ $\quad(\partial f_\sigma \leq q-2)$
- $N_1(q) = q(q-1)$
- $N_d(q) = 0$ if $d | q-1$ (Hermite criterion)
- $N_d(q)$ is known for $d < 6$
-

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- $\sum_{d \leq q-2} N_d(q) = q!$ $(\partial f_\sigma \leq q-2)$
- $N_1(q) = q(q-1)$
- $N_d(q) = 0$ if $d | q-1$ (Hermite criterion)
- $N_d(q)$ is known for $d < 6$
- Almost all permutation polynomials have degree $q-2$

Enumeration of PP by degree

$$N_d(q) = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) = d\}$$

Problem. Compute $N_d(q)$

- $\sum_{d \leq q-2} N_d(q) = q!$ $(\partial f_\sigma \leq q-2)$
- $N_1(q) = q(q-1)$
- $N_d(q) = 0$ if $d|q-1$ (Hermite criterion)
- $N_d(q)$ is known for $d < 6$
- Almost all permutation polynomials have degree $q-2$

(S. Konyagin, FP – 2002) $M_q = \{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial f_\sigma < q-2\}$

$$|\#M_q - (q-1)!| \leq \sqrt{2e/\pi}\, q^{q/2}$$

A recent result

A recent result

$$\mathcal{N}_d = \#\left\{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) < q - d - 1\right\}$$

A recent result

$$\mathcal{N}_d = \#\{\sigma \in \mathcal{S}(\mathbb{F}_q) \mid \partial(f_\sigma) < q - d - 1\}$$

Theorem *S. Konyagin, FP – 2003*

Let $\alpha = (e-2)/3e = 0.08808\cdots$ *and* $d < \alpha q$. *Then*

$$\left|\mathcal{N}_d - \frac{q!}{q^d}\right| \leq 2^d d q^{2+q-d} \binom{q}{d} \left(\frac{2d}{q-d}\right)^{(q-d)/2}.$$

It follows that

$$\mathcal{N}_d \sim \frac{q!}{q^d}$$

if $d \leq \alpha q$ *and* $\alpha < 0.03983$

Other ways of counting

If $\sigma \in \mathcal{S}(\mathbb{F}_q)$, $\boxed{c_\sigma = \#\{a \in \mathbb{F}_q \mid \sigma(a) \neq a\}}$

$$\sigma \neq id \Rightarrow \quad q - c_\sigma \leq \partial f_\sigma \leq q - 2$$

(since $f_\sigma(x) - x$ has at least $q - c_\sigma$ roots)

Consequences.

- 2–cycles have degree $q-2$
- 3–cycles have degree $q-2$ or $q-3$
- k–cycles have degree in $[q-k, q-2]$

(Wells)
$$\boxed{\#\{\sigma \in 3\text{–cyle},\ \partial(f_\sigma) = q-3\} = \begin{cases} \frac{2}{3}q(q-1) & q \equiv 1 \bmod 3 \\ 0 & q \equiv 0 \bmod 3 \\ \frac{1}{3}q(q-1) & q \equiv 0 \bmod 3 \end{cases}}$$

More enumeration functions

- σ_1, σ_2 conjugated $\Longrightarrow$ $c_{\sigma_1} = c_{\sigma_2}$
- $\mathcal{C}$ conjugation class of permutations
- $c_{\mathcal{C}} = \#\{$ elements $\in \mathbb{F}_q$ moved by any $\sigma \in \mathcal{C}\}$
  (i.e. $c_{\mathcal{C}} = c_\sigma$ for any $\sigma \in \mathcal{C}$ $\quad q - c_{\mathcal{C}} \leq f_\sigma$)
- $\mathcal{C} = [k]$=k–cycles $\Longrightarrow$ $c_{[k]} = k$
- Natural enumeration functions:
  - ✗ $m_{\mathcal{C}}(q) = \#\{\sigma \in \mathcal{C}, \partial f_\sigma = q - c_{\mathcal{C}}\}$ (minimal degree)
  - ✗ $M_{\mathcal{C}}(q) = \#\{\sigma \in \mathcal{C}, \partial f_\sigma < q - 2\}$ (non-maximal degree)

Permutation Classes with non maximal degree

Let $\mathcal{C} = (m_1, \ldots, m_t)$ be the class of permutations with m_1 1-cycles, ..., m_t t-cycles. The number $c_{\mathcal{C}}$ of elements in $\mathbb{F}_q$ moved by any element of $\mathcal{C}$ is

$$c_{\mathcal{C}} = 2m_2 + 3m_3 + \cdots + tm_t$$

$$\boxed{M_{\mathcal{C}}(q) = \#\{\sigma \ \in \mathcal{C}, \partial f_\sigma < q-2\}}$$

Theorem 1 (C. Malvenuto, FP - 2002). $\exists N = N_{\mathcal{C}} \in \mathbb{N}$, $f_1, \cdots, f_N \in \mathbb{Z}[x]$, f_i monic, $\partial f_i = c_{\mathcal{C}} - 3$ such that if $q \equiv a \bmod N$, then

$$M_{\mathcal{C}}(q) = \frac{q(q-1)}{m_2! 2^{m_2} \cdots m_t! t^{m_t}} f_a(q)$$

k–cycles with minimal degree

$$m_{[k]}(q) = \#\{\sigma\ k\text{–cycle}, \partial f_\sigma = q - k\}$$

THEOREM 2 (C. Malvenuto, FP - 2003).

- If $q \equiv 1 \bmod k \quad \Longrightarrow$

$$m_{[k]}(q) \geq \frac{\varphi(k)}{k} q(q-1).$$

- If $q = p^f$, $p \geq 2 \cdot 3^{[k/3]-1} \quad \Longrightarrow$

$$m_{[k]}(q) \leq \frac{(k-1)!}{k} q(q-1).$$

Consequences of Theorem 1

- $\dfrac{M_{\mathcal{C}}(q)}{\#\mathcal{C}} = \dfrac{1}{q} + O\left(\dfrac{1}{q^2}\right)$
- If $\mathcal{C}$ is fixed,

$$\mathrm{Prob}(\partial f_\sigma < q - 2 \mid \sigma \in \mathcal{C}) \sim \frac{1}{q}$$

- If $q = 2^r$, $\mathcal{C}_r$ is the conjugation class of r transposition,

$$M_{\mathcal{C}_r}(q) = \frac{q!}{r!2^r(q-2r+1)!} - \frac{q-2(r-1)(2r-1)}{2r} M_{\mathcal{C}_{r-1}}(q)$$

- One can compute $M_{\mathcal{C}}(q)$ for $c_{\mathcal{C}} \leq 6$

Table 1. $\#c_{\mathcal{C}} \leq 6$, (q odd)

- $M_{[4]}(q) = \frac{1}{4}q(q-1)\left(q-5-2\eta(-1)-4\eta(-3)\right)$

- $M_{[2\ 2]}(q) = \frac{1}{8}q(q-1)(q-4)\left\{1+\eta(-1)\right\}$

- $$M_{[5]}(q) = \frac{1}{5}q(q-1)\left(q^2-(9-\eta(5)-5\eta(-1)+5\eta(-9))\,q+ +26+5\eta(-7)+15\eta(-3)+15\eta(-1)\right)$$

- $$M_{[2\ 3]}(q) = \frac{1}{6}q(q-1)\left(q^2-(9+\eta(-3)+3\eta(-1))q+ +(24+6\eta(-3)+18\eta(-1)+6\eta(-7))\right)+ \eta(-1)(1-\eta(9))q(q-5).$$

Table 2. $\#c_{\mathcal{C}} \leq 6$, (q even)

- $M_{[4]}(2^n) = \frac{1}{4}2^n(2^n-1)(2^n-4)(1+(-1)^n)$
- $M_{[2\ 2]}(2^n) = \frac{1}{8}2^n(2^n-1)(2^n-2)$
- $M_{[5]}(2^n) = \frac{1}{5}2^n(2^n-1)(2^n-3-(-1)^n)(2^n-6-3(-1)^n)$
- $M_{[2\ 3]}(2^n) = \frac{1}{6}2^n(2^n-1)(2^n-3-(-1)^n)(2^n-6)$.

Table 3. $\#c_{\mathcal{C}} = 6$, (q odd, $3 \nmid q$)

$$
\begin{aligned}
M_{[6]}(q) &= \tfrac{q(q-1)}{6}\{q^3 - 14\,q^2 + [68 - 6\,\eta(5) - 6\,\eta(50)]q - \\
&\quad [154 + 66\,\eta(-3) + 93\,\eta(-1) + 12\,\eta(-2) + 54\,\eta(-7)]\} \\
M_{[4\ 2]}(q) &= \tfrac{q(q-1)}{8}(q^3 - [14 - \eta(2)]q^2 + \\
&\quad [71 + 12\,\eta(-1) + \eta(-2) + 4\,\eta(-3) - 8\,\eta(50)]q \\
&\quad -[148 + 100\,\eta(-1) + 24\,\eta(-2) + 44\,\eta(-3) + 40\,\eta(-7)]) \\
M_{[3\ 3]}(q) &= \tfrac{q(q-1)}{18}(q^3 - 13\,q^2 + [62 + 9\,\eta(-1) + 4\,\eta(-3)]q \\
&\quad -[150 + 99\,\eta(-1) + 42\,\eta(-3) + 72\,\eta(-7)]) \\
M_{[2\ 2\ 2]}(q) &= \tfrac{q(q-1)}{48}(q^3 - [14 + 3\,\eta(-1)]q^2 + [70 + 36\,\eta(-1) + 6\,\eta(-2)]q \\
&\quad -[136 + 120\,\eta(-1) + 48\,\eta(-2) + 8\,\eta(-3)])
\end{aligned}
$$

Table 4. $\#c_{\mathcal{C}} = 6$

$$M_{[6]}(3^n) = \frac{3^n(3^n-1)}{6}\{3^{3n} - [14+2(-1)^n]3^{2n} + [71+39(-1)^n]3^n - [162+147(-1)^n]\}$$

$$M_{[4\ 2]}(3^n) = \frac{3^n(3^n-1)}{8}\{3^{3n} - [14+3\,(-1)^n]3^{2n} + [72+40\,(-1)^n]3^n - [164+140\,(-1)^n]\}$$

$$M_{[3\ 3]}(3^n) = \frac{3^n(3^n-1)}{18}\{(1+(-1)^n)\,3^{3\,n} - [14+15\;(-1)^n]3^{2\,n} + [71+81\;(-1)^n]3^n - [150+171\;(-1)^n]\}$$

$$M_{[2\ 2\ 2]}(3^n) = \frac{3^n(3^n-1)}{48}\{3^{3n} - [14+3(-1)^n]3^{2n} + [76+36(-1)^n]3^n - +[168+120(-1)^n]\}$$

Table 5. $\#c_{\mathcal{C}} = 6$

$$M_{[6]}(2^n) = \frac{2^n(2^n-1)}{6} \{(2^n - 3 - (-1)^n)(2^{2n} - (11 - (-1)^n)2^n + (41 + 7(-1)^n))\}$$

$$M_{[4\ 2]}(2^n) = \frac{2^n(2^n-1)}{8} \{(2^n - 3 - (-1)^n)(2^{2n} - 11 \cdot 2^n + 37 + (-1)^n)\}$$

$$M_{[3\ 3]}(2^n) = \frac{2^n(2^n-1)}{18} \{(2^n - 3 - (-1)^n)(2^{2n} - (10 - (-1)^n)2^n + 45 - 3(-1)^n))\}$$

$$M_{[2\ 2\ 2]}(2^n) = \frac{2^n(2^n-1)}{48} \{(2^n - 2)(2^n - 4)(2^n - 8)\}$$

Sketch of the Proof of Theorem 2. (1/3)

STEP 1. Translate the problem into one on counting points of an algebraic varieties

$$m_k(q) = \frac{q(q-1)}{k} n_k(q)$$

where $n_k(q) = \{\sigma \in [k] \mid \partial f_\sigma = q - k, \sigma(0) = 1\}$.
Need to show $|n_k(q)| \leq (k-1)!$. Now

$$f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c) \left(1 - (x-c)^{q-1}\right) = A_1 x^{q-2} + A_2 x^{q-3} + \cdots + A_{q-1}.$$

with $A_j = \sum_{c \in \mathbb{F}_q} \sigma(c) c^j = \sum_{c \in \mathbb{F}_q} \sigma(c) \left(c^j - c^{j-1}\right) = \sum_{\substack{c \in \mathbb{F}_q \\ \sigma(c) \neq c}} (\sigma(c) - c) c^j.$

Sketch of the Proof of Theorem 2. (2/3)

If $\sigma = (0,\ 1,\ x_1,\ x_2,\ \ldots,\ x_{k-2}) \in \mathcal{S}(\mathbb{F}_q)$,

$$A_j(\sigma) = (1 - x_1) + (x_1 - x_2)x_1^j + \cdots (x_{k-2} - x_{k-2})x_{k-3}^j + x_{k-2}^{j+1}.$$

Def. (Affine k–th Silvia set)

$$\mathcal{A}_k : \begin{cases} (1 - x_1) + x_1(x_1 - x_2) + \cdots + x_{k-3}(x_{k-3} - x_{k-2}) + x_{k-2}^2 & = \quad 0 \\ (1 - x_1) + x_1^2(x_1 - x_2) + \cdots + x_{k-3}^2(x_{k-3} - x_{k-2}) + x_{k-2}^3 & = \quad 0 \\ & \vdots \\ (1 - x_1) + x_1^{k-2}(x_1 - x_2) + \cdots + x_{k-3}^{k-2}(x_{k-3} - x_{k-2}) + x_{k-2}^{k-1} & = \quad 0 \end{cases}$$

$$n_k(q) = \#\{\underline{x} = (x_1, \ldots, x_{k-2}) \in \mathbb{F}_q^{k-2} \mid \underline{x} \in \mathcal{A}_k(\mathbb{F}_q), x_i \neq x_j\} \leq \#\mathcal{A}_k(\mathbb{F}_q)$$

$$\boxed{\dim_{\overline{\mathbb{F}}_q} \mathcal{A}_k = 0 \quad \overset{\text{Bezout Thm.}}{\Longrightarrow} \quad \#\mathcal{A}(\mathbb{F}_q) \leq (k-1)!}$$

Sketch of the Proof of Theorem 2. (3/3)

STEP 2.

Theorem. If $\mathbf{K}$ is an algebrically closed field,

$$\mathrm{char}(\mathbf{K}) = \begin{cases} 0 & \text{or} \\ > 2 \cdot 3^{[k/3]-1}. \end{cases}$$

Then

$$\boxed{\dim_{\mathbf{K}} \mathcal{A}_k = 0.}$$

NOTE.

- Proof is based on finding projective hyperplanes disjoint from $\mathcal{A}_k$
- There are examples of small values of q with $\dim_{\mathbf{K}} \mathcal{A}_k > 0$