Measuring small subgroup attacks against Diffie-Hellman

Luke Valenta[*], David Adrian[†], Antonio Sanso[‡], Shaanan Cohney[*], Joshua Fried[*], Marcella Hastings[*], J. Alex Halderman[†], Nadia Heninger[*]

[*]University of Pennsylvania
[†]University of Michigan
[‡]Adobe

February 28, 2017

This work

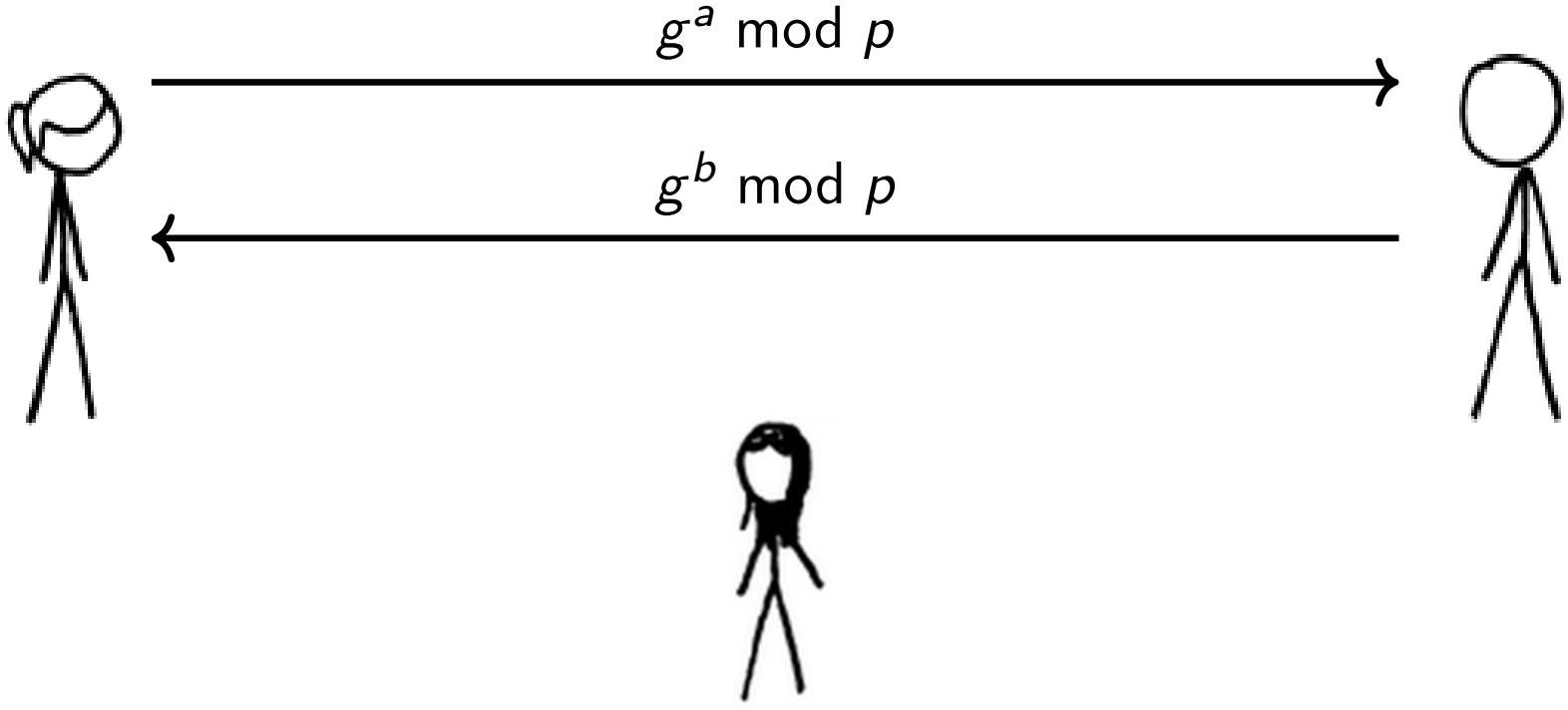
- Revisit decades-old small subgroup attacks in Diffie-Hellman
- Looked at hosts and implementations in the wild
- Punch line: Nobody implements the countermeasures!
- Emerged from **Logjam** [ABDGGHHSTVVWZZ 2015]

Textbook (Finite-Field) Diffie-Hellman Key Exchange

[Diffie Hellman 1976]

p a prime (so $\mathbb{F}_p^*$ is a cyclic group)

$g < p$ group generator (often 2 or 5)



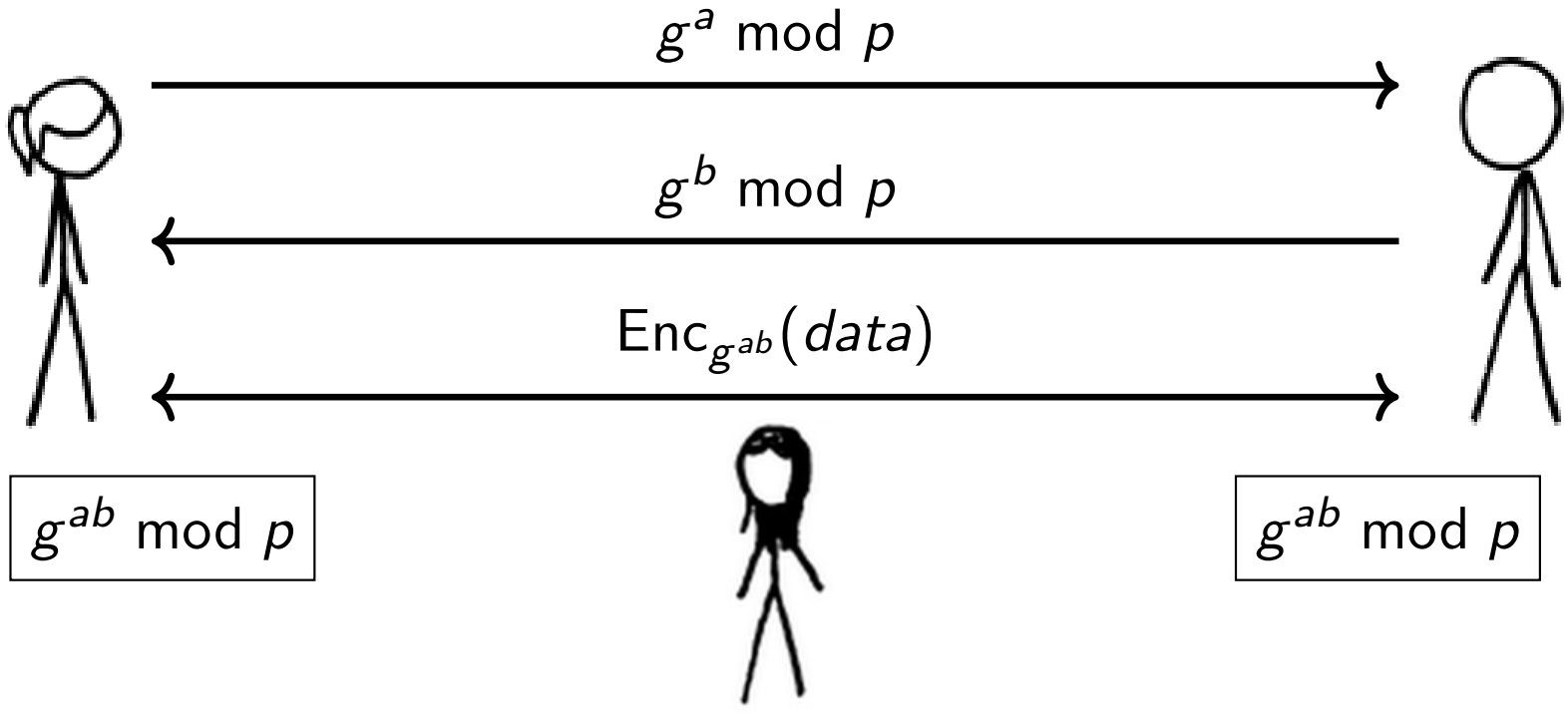
Images from XKCD

Textbook (Finite-Field) Diffie-Hellman Key Exchange

[Diffie Hellman 1976]

p a prime (so $\mathbb{F}_p^*$ is a cyclic group)

$g < p$ group generator (often 2 or 5)



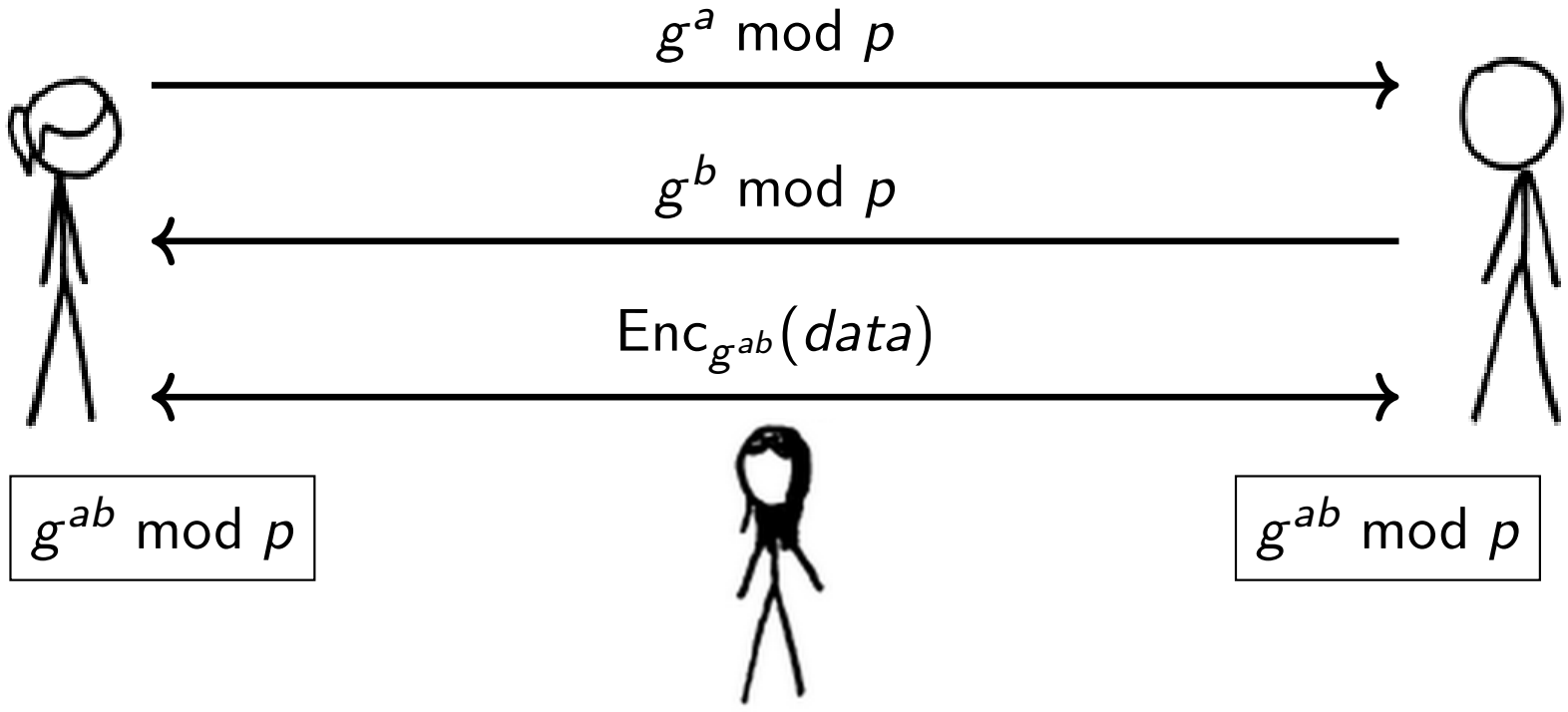
Images from XKCD

Textbook (Finite-Field) Diffie-Hellman Key Exchange

[Diffie Hellman 1976]

p a prime (so $\mathbb{F}_p^*$ is a cyclic group)

$g < p$ group generator (often 2 or 5)



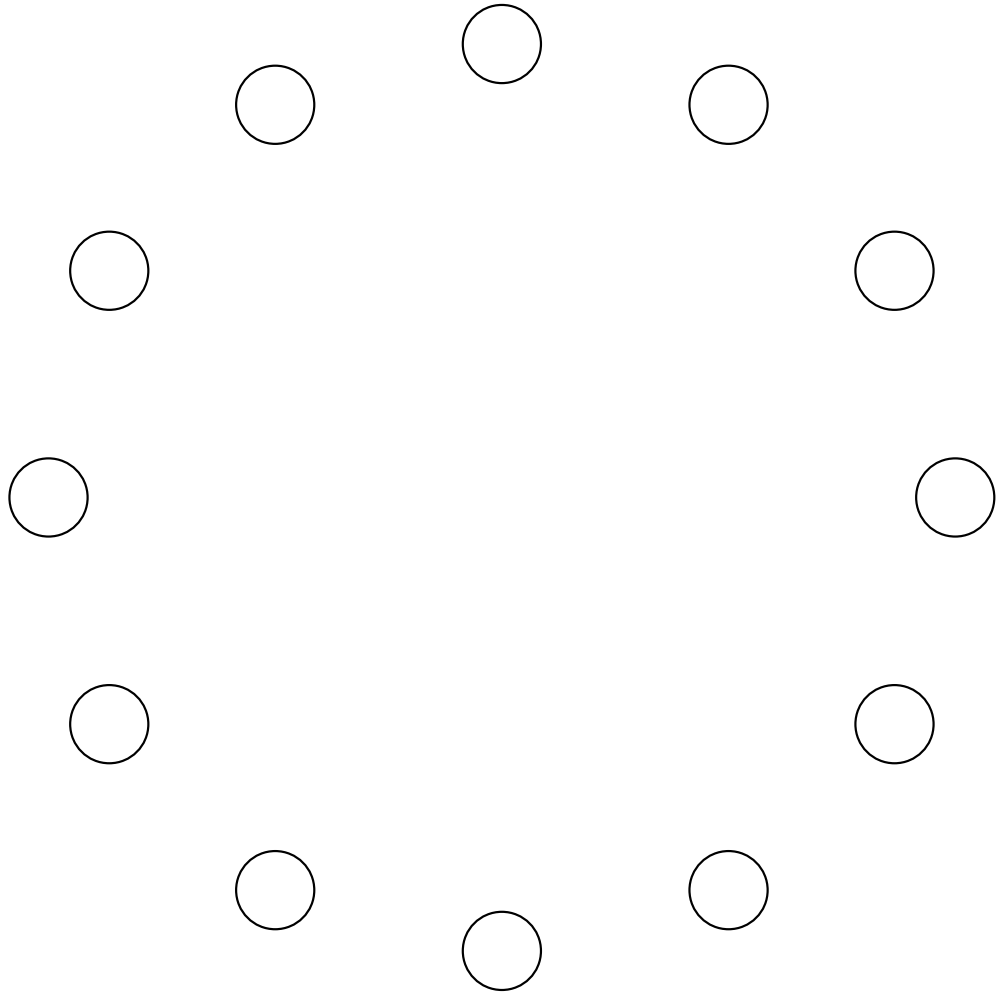
Images from XKCD

NH: "There are dragons swimming under the placid surface of this beautiful mathematical lake."

Background: groups, subgroups, and generators

Cyclic group

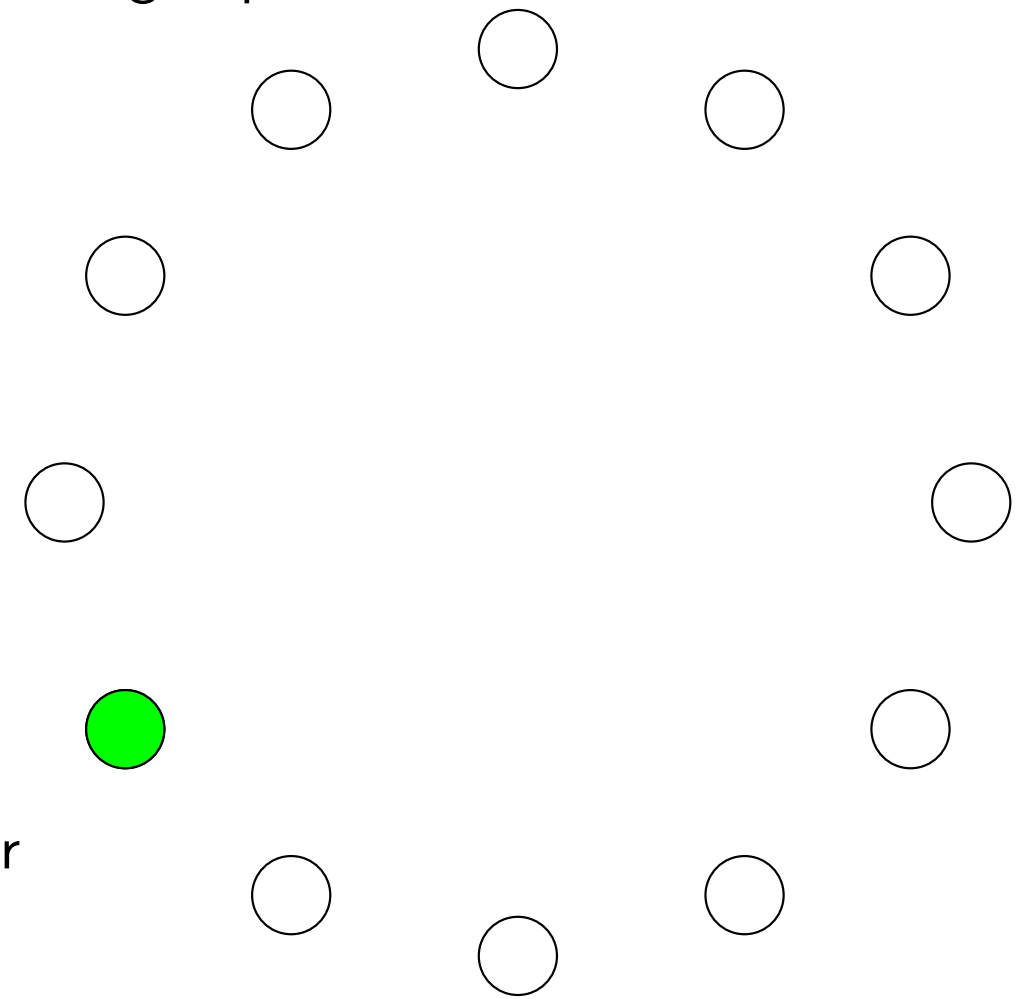
Order = #elements in group



Background: groups, subgroups, and generators

Cyclic group

Order = #elements in group

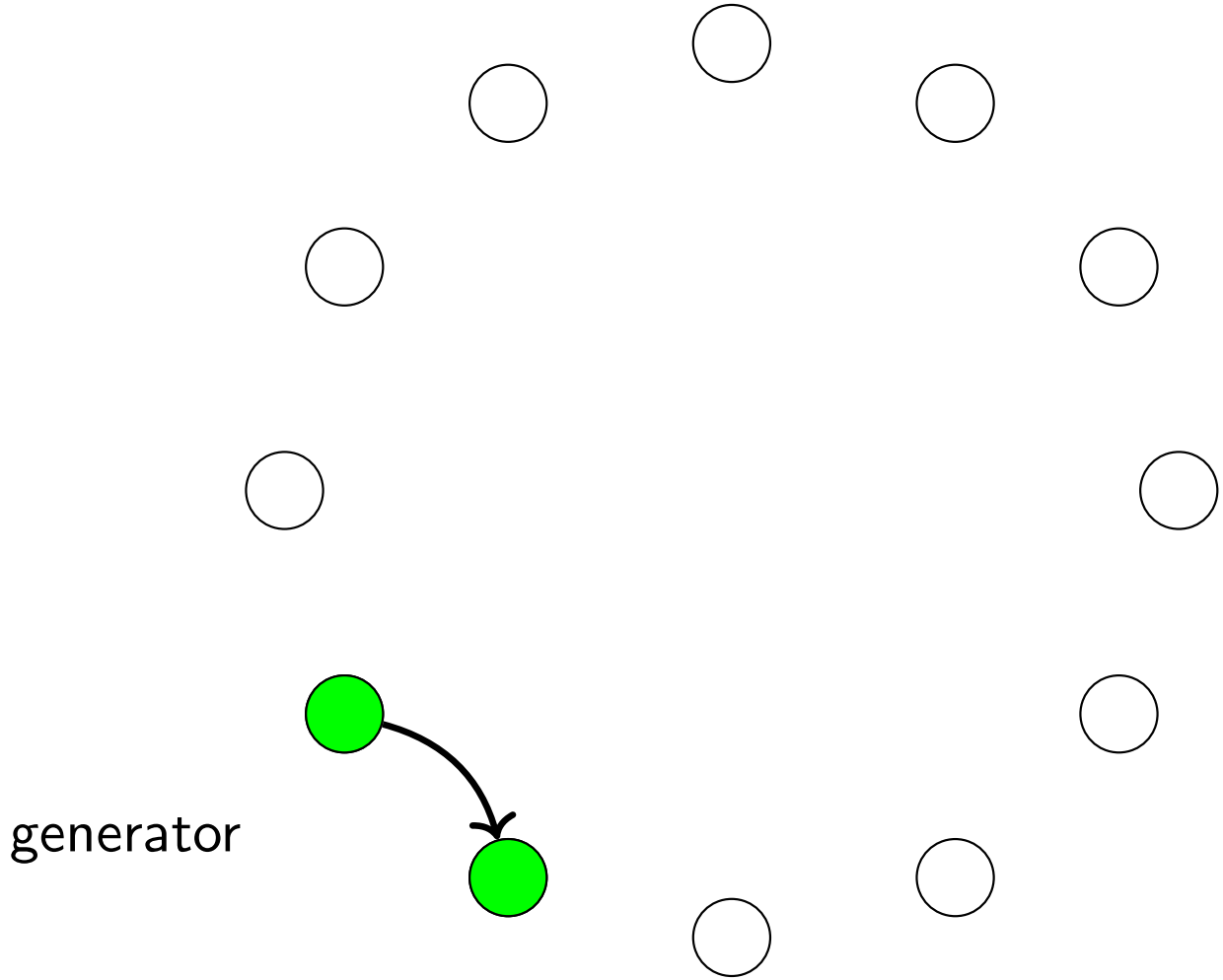


generator

Background: groups, subgroups, and generators

Cyclic group

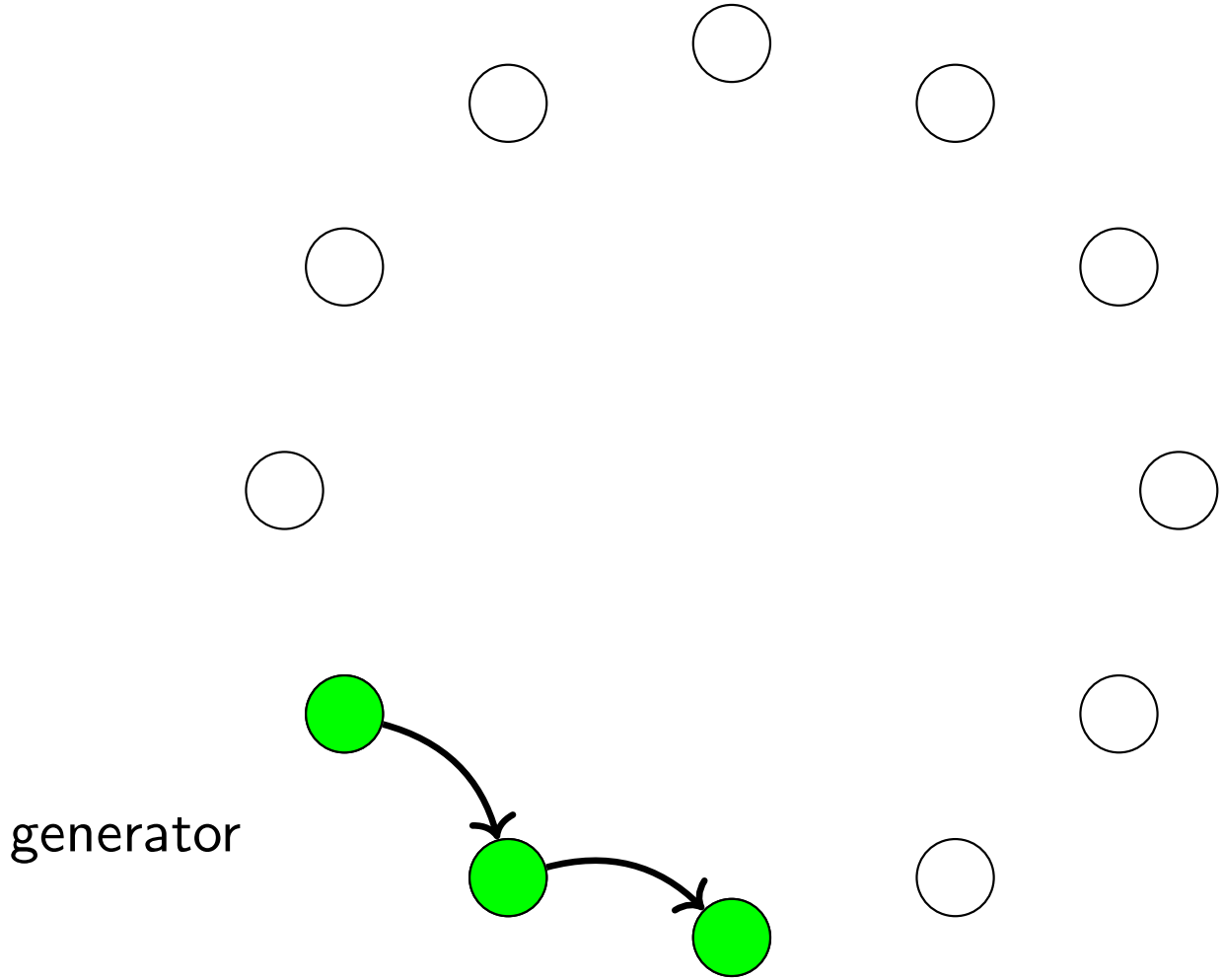
Order = #elements in group



Background: groups, subgroups, and generators

Cyclic group

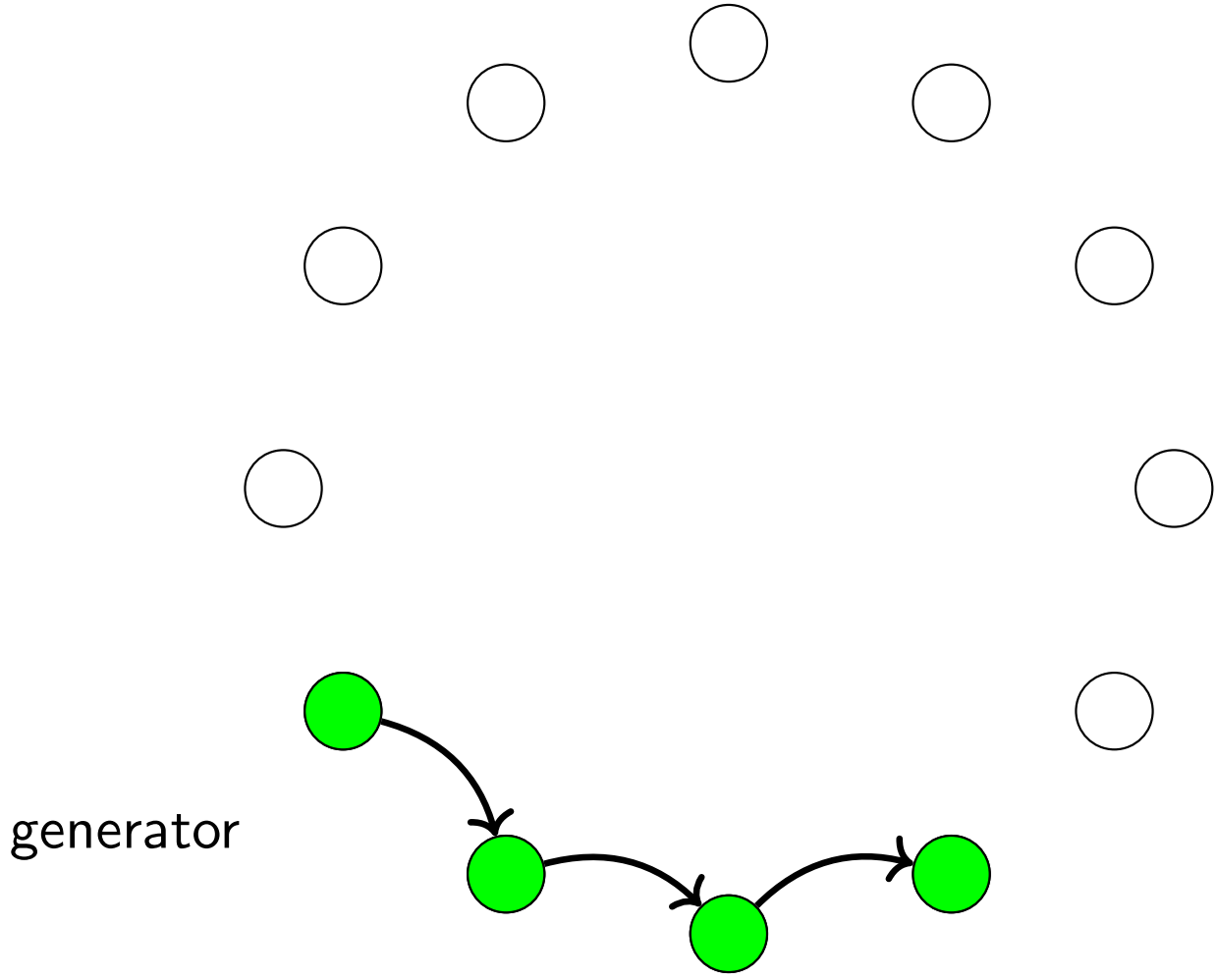
Order = #elements in group



Background: groups, subgroups, and generators

Cyclic group

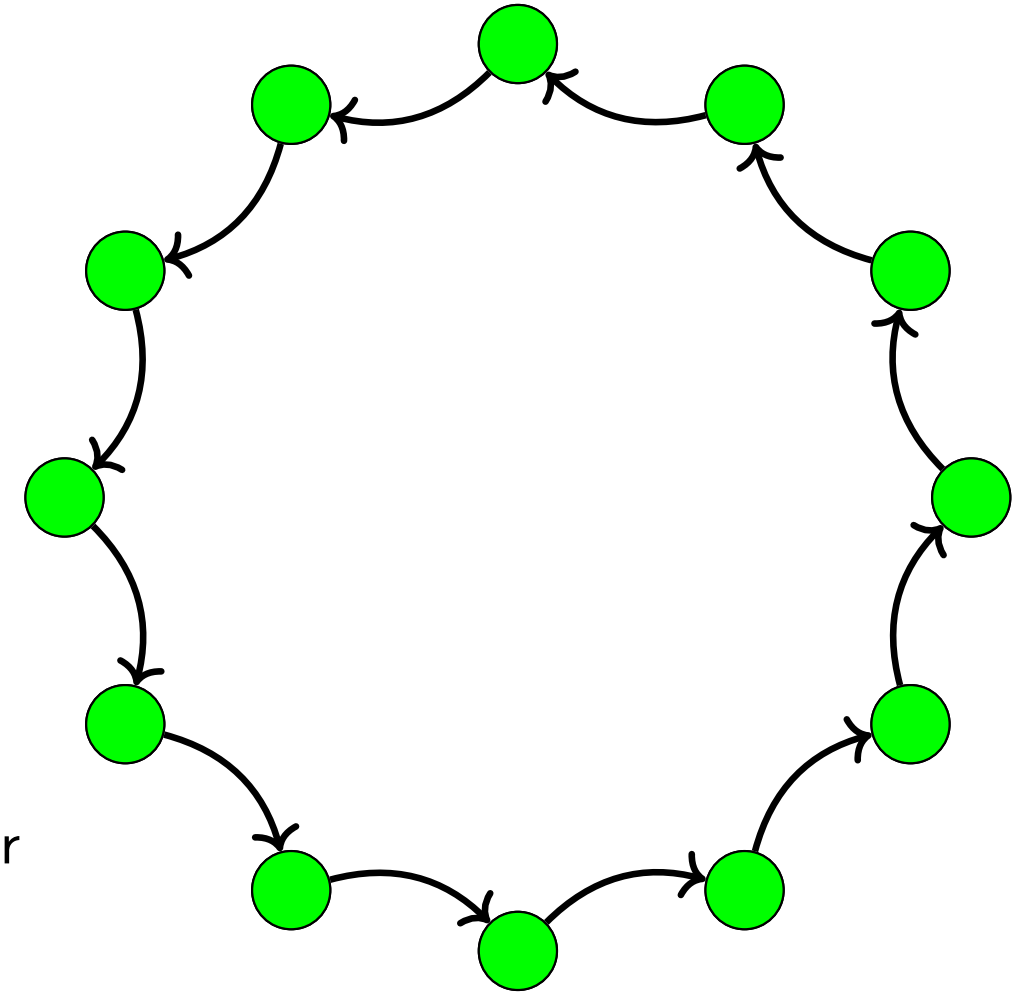
Order = #elements in group



Background: groups, subgroups, and generators

Cyclic group

Order = #elements in group



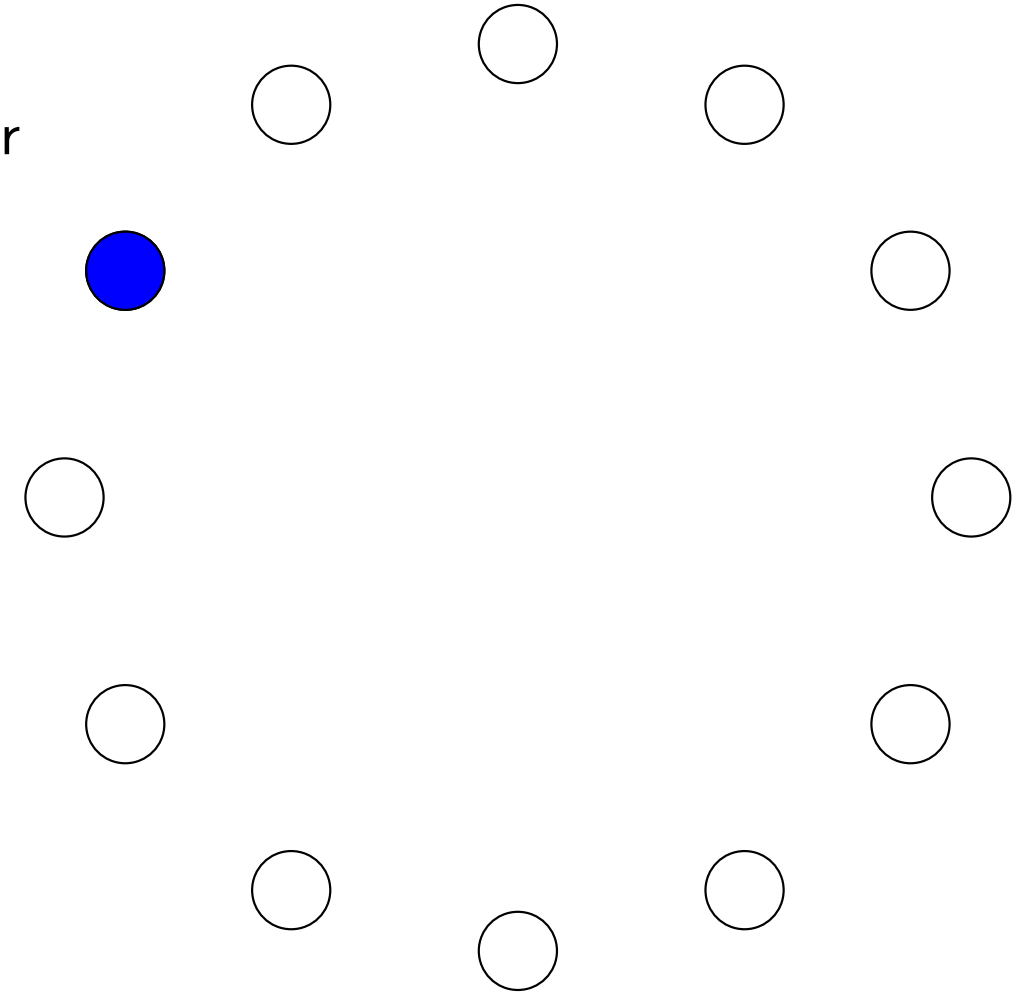
generator

Background: groups, subgroups, and generators

Subgroup

Order = #elements in subgroup

generator

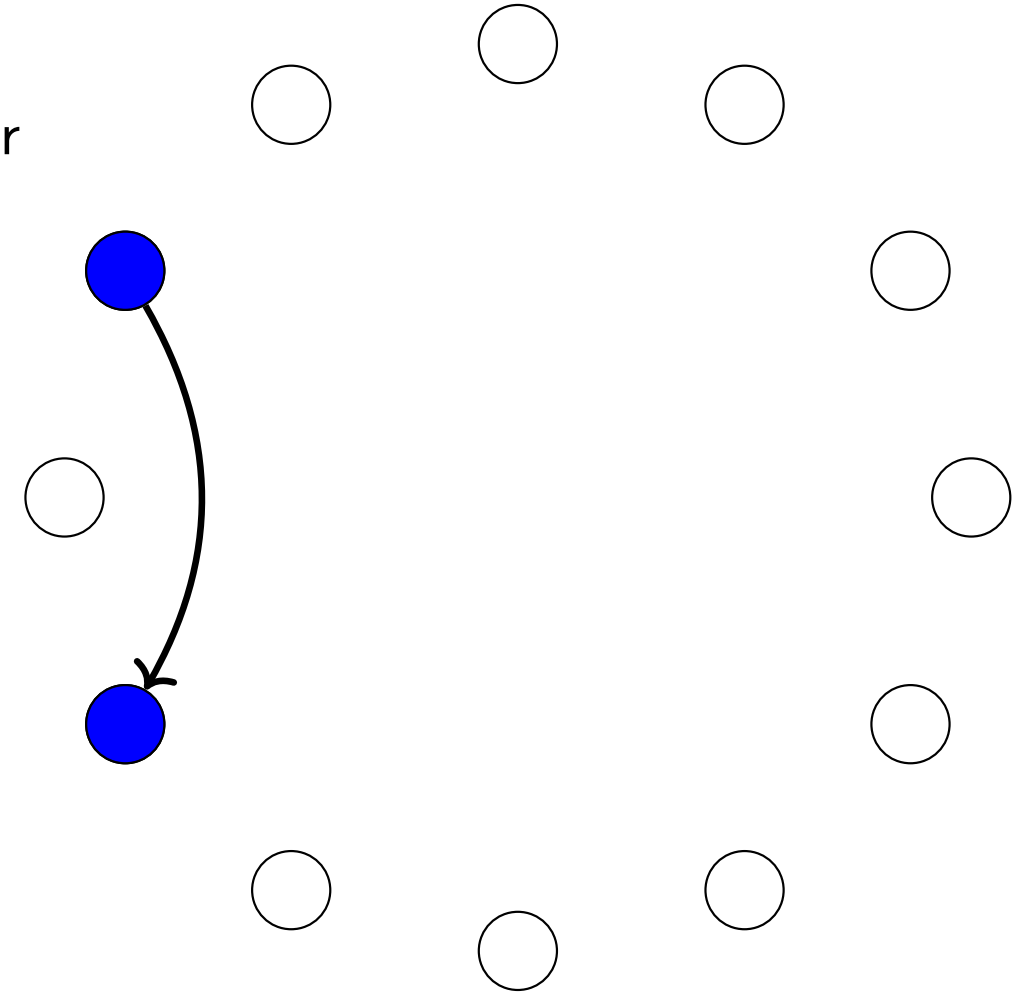


Background: groups, subgroups, and generators

Subgroup

Order = #elements in subgroup

generator

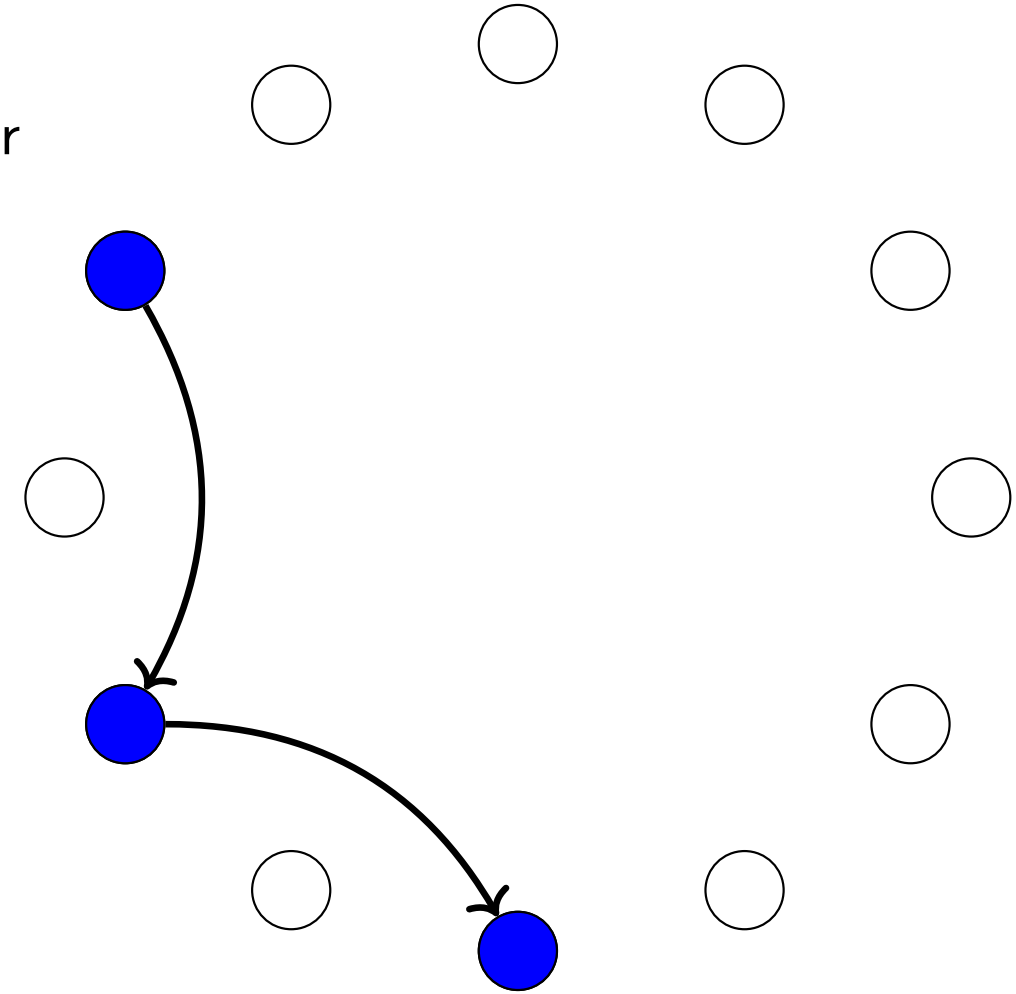


Background: groups, subgroups, and generators

Subgroup

Order = #elements in subgroup

generator

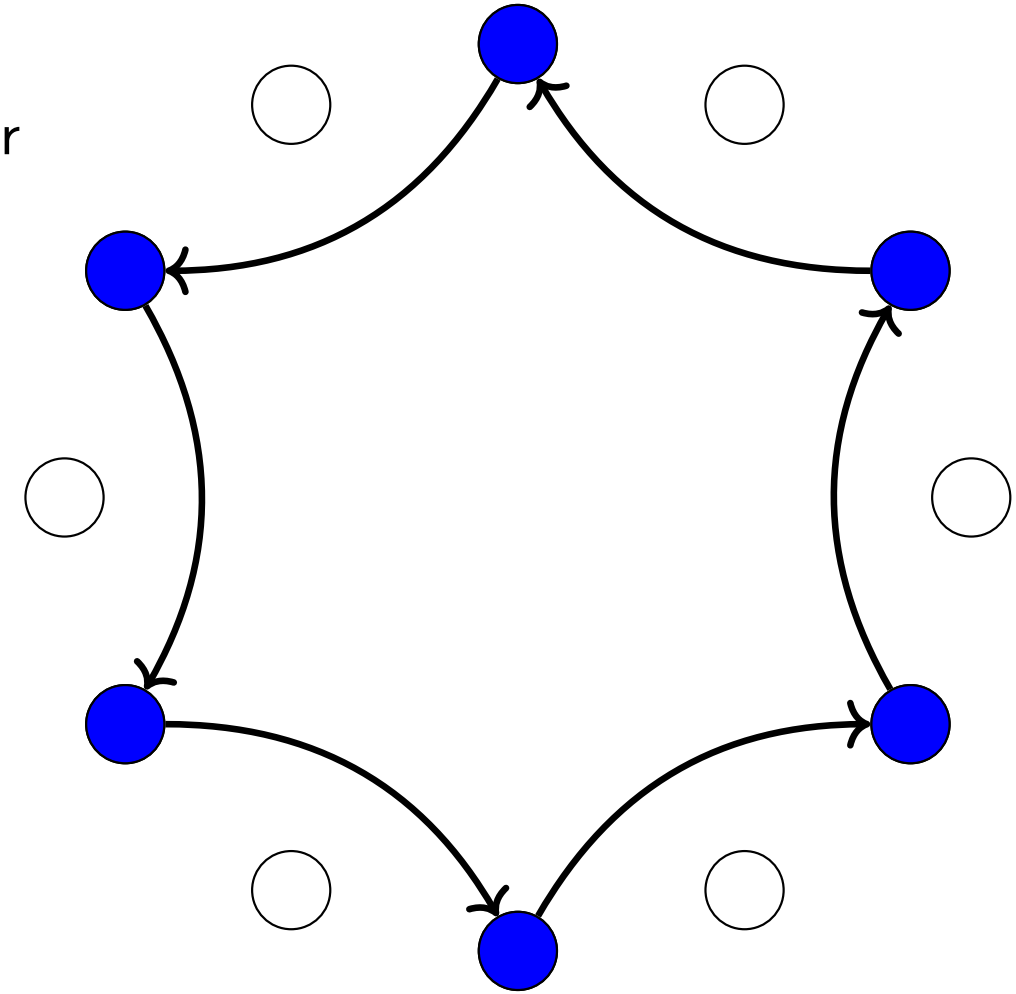


Background: groups, subgroups, and generators

Subgroup

Order = #elements in subgroup

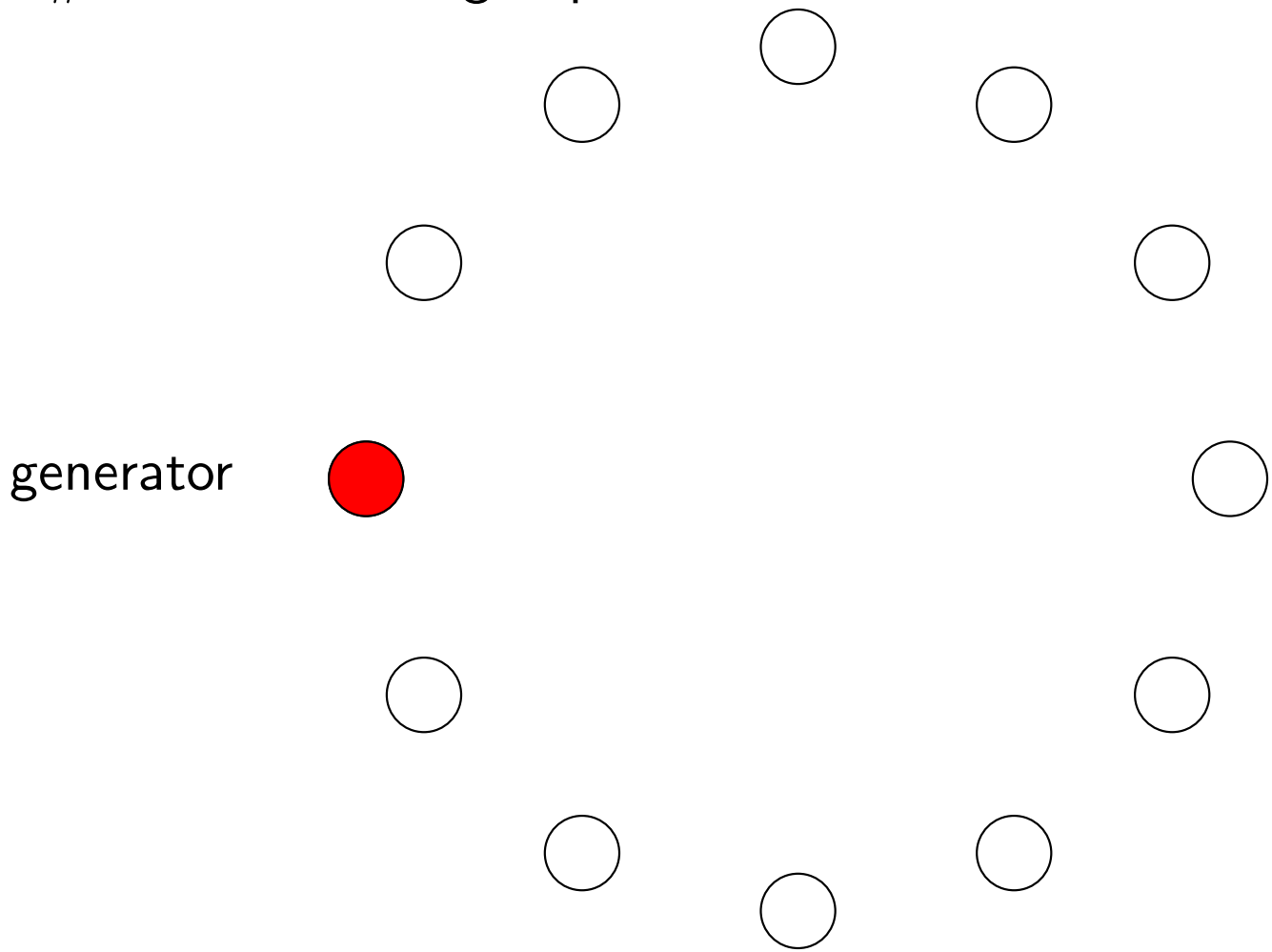
generator



Background: groups, subgroups, and generators

Small subgroup

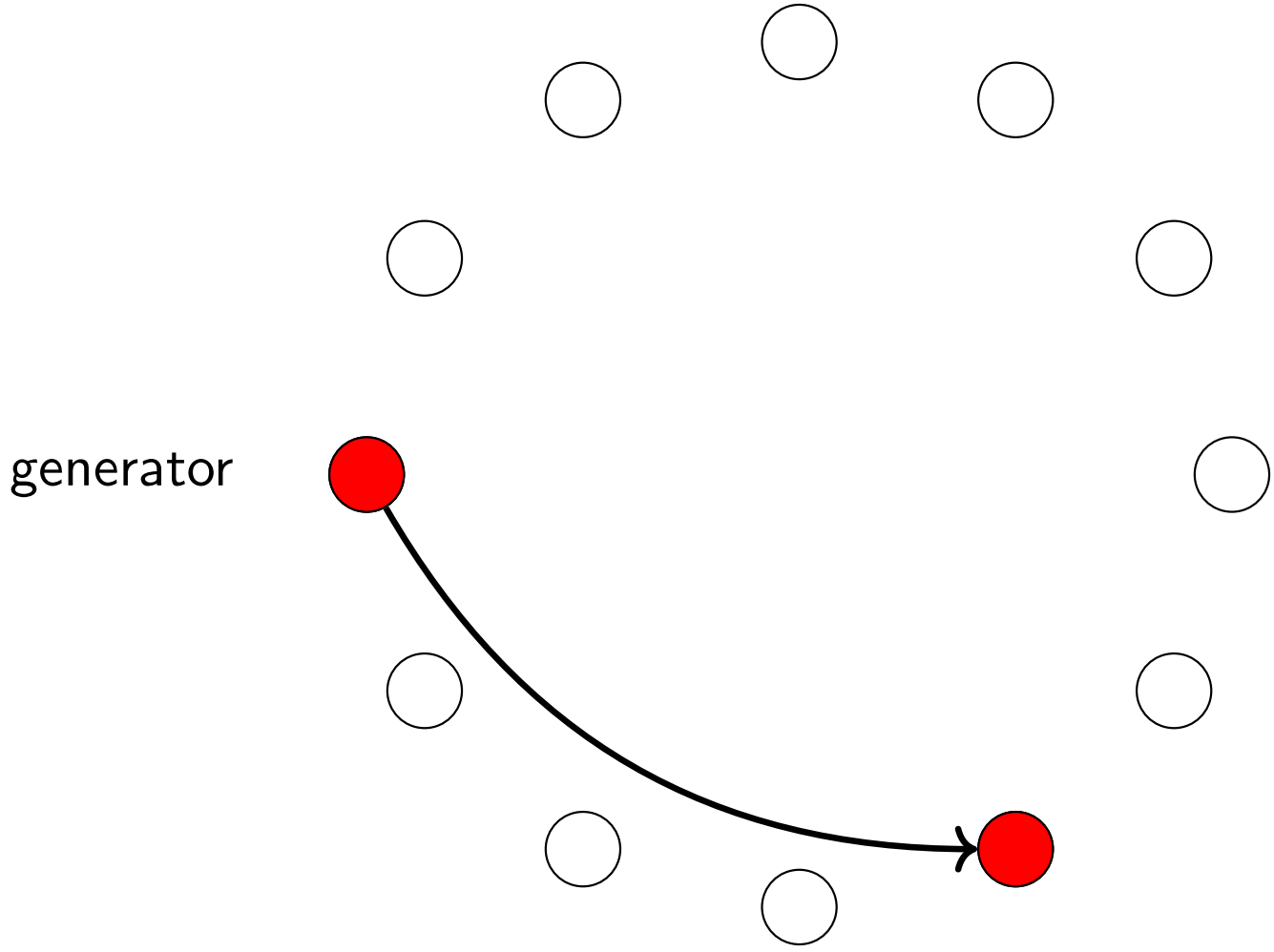
Order = #elements in subgroup



Background: groups, subgroups, and generators

Small subgroup

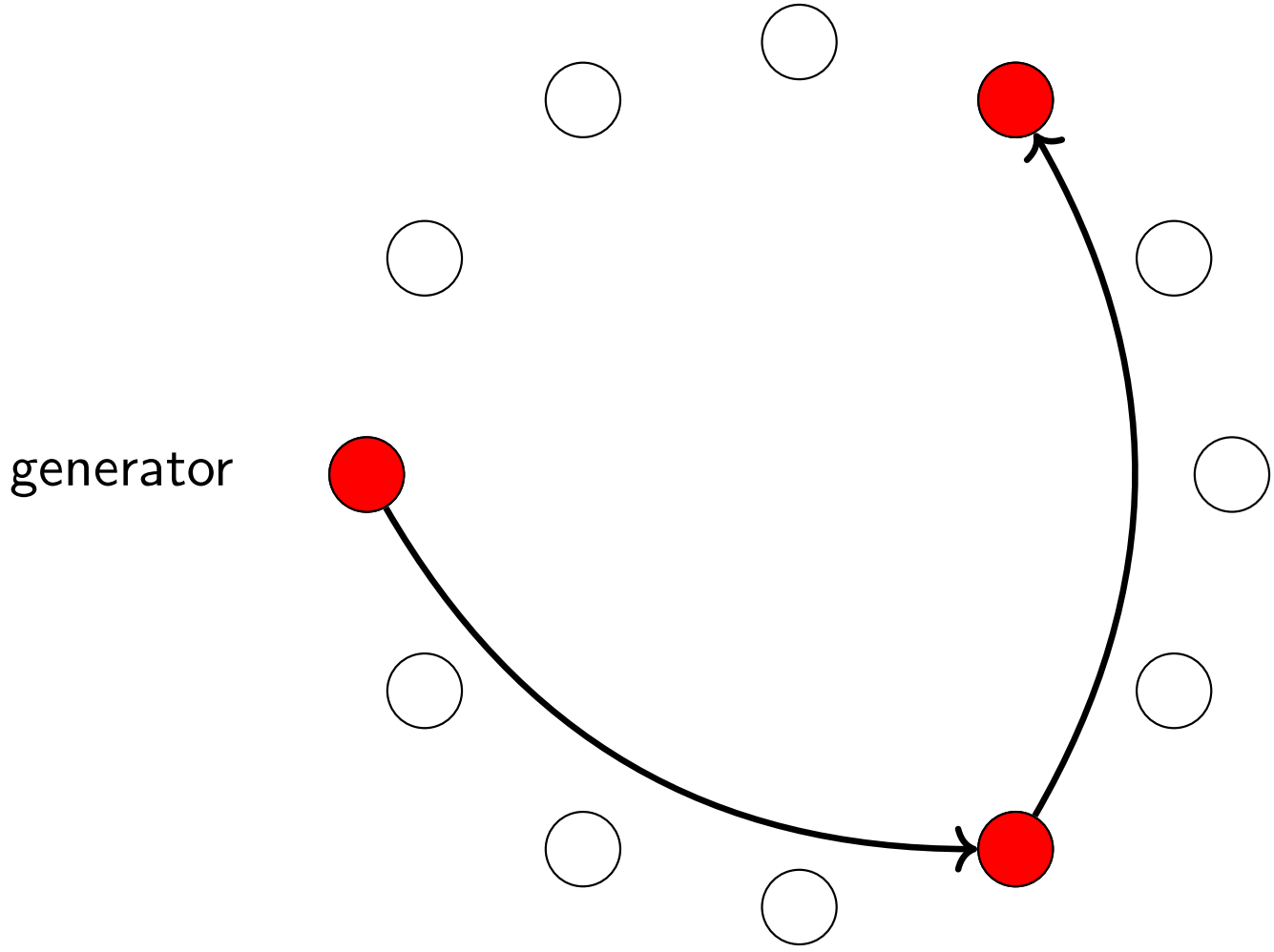
Order = #elements in subgroup



Background: groups, subgroups, and generators

Small subgroup

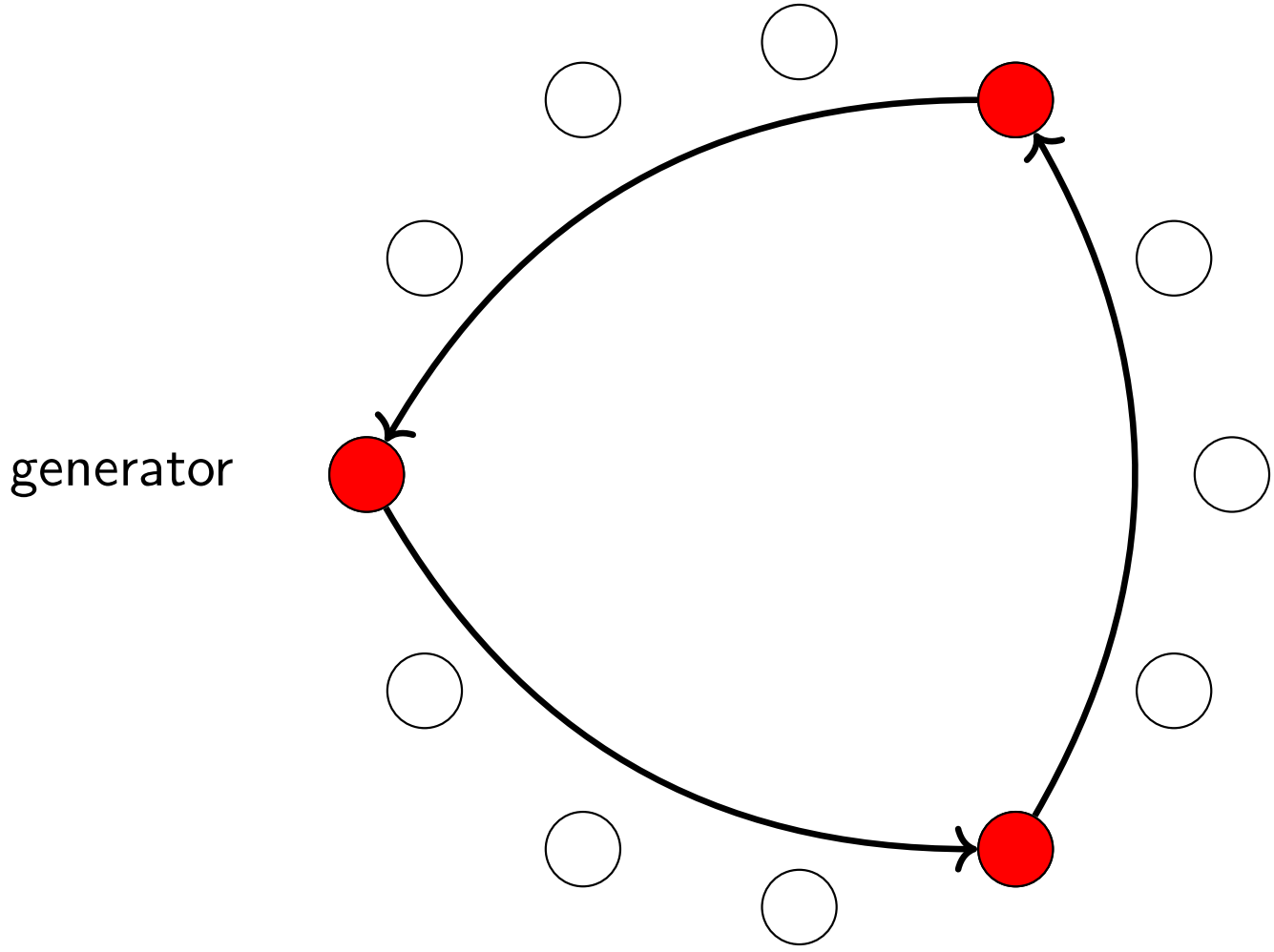
Order = #elements in subgroup



Background: groups, subgroups, and generators

Small subgroup

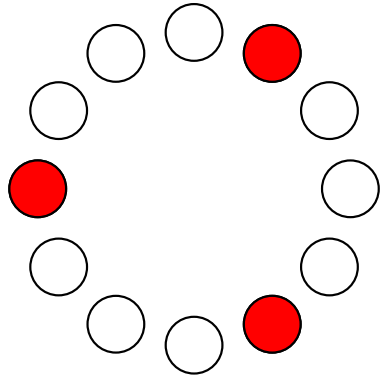
Order = #elements in subgroup



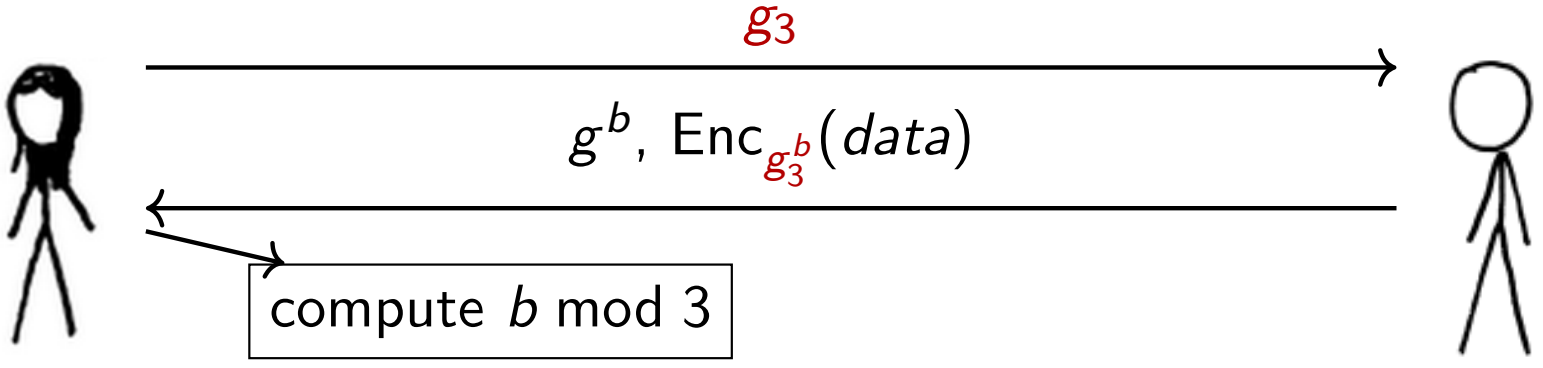
Existence of small subgroups $\rightarrow$ small subgroup attacks.

g generates correct subgroup of order q

g_3 generates subgroup of order 3



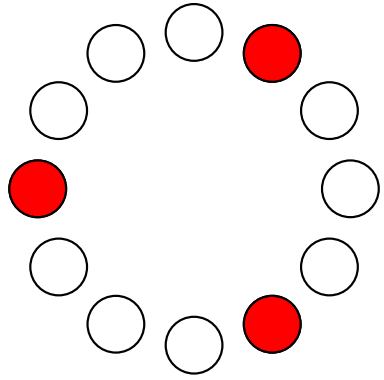
[Lim Lee 1997]



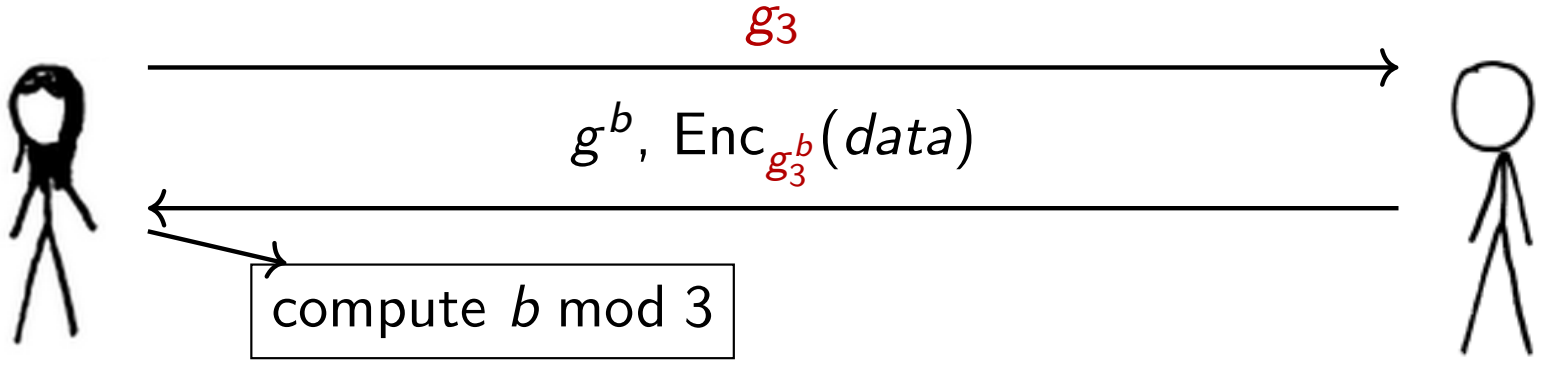
Existence of small subgroups $\rightarrow$ small subgroup attacks.

g generates correct subgroup of order q

g_3 generates subgroup of order 3



[Lim Lee 1997]



Repeat for many small subgroups $\Longrightarrow$ find b using Chinese Remainder Theorem

Small subgroup attacks

Made much worse with...

- Many small subgroups (i.e., p-1 has many small factors)
- Short secret exponents (common optimization)
- Reused Diffie-Hellman values (common optimization)

Countermeasures

The countermeasures against these attacks are well known, and built into every DH standard:

- Use a "safe" prime $p = 2q + 1$, where q is prime
  1. Verify $2 \leq y \leq p - 2$ (otherwise, may leak 1 bit)
- Use a subgroup of large prime order q mod p
  1. Verify $2 \leq y \leq p - 2$
  2. Verify $1 = y^q \bmod p$

Inspiration for our work

The attacks and defenses are known. Why is this work interesting?

Inspiration for our work

The attacks and defenses are known. Why is this work interesting?

"The Internet is vast, and filled with bugs."

—Adam Langley, Crypto 2013

Inspiration for our work

The attacks and defenses are known. Why is this work interesting?

"The Internet is vast, and filled with bugs."

—Adam Langley, Crypto 2013

Theorem (Murphy's law)

Anything that can go wrong, will go wrong.

Corollary

If it is possible for an implementation to have made a mistake, someone has.

Standards mandate smaller subgroups

Leaves room for implementation mistakes

NIST SP800-56a: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography

Table 1: FFC parameter-size sets

FFC parameter-size set name	FA	FB	FC
Maximum security strength supported (in bits)	80	112	112
Bit length of field size p (i.e., $\lceil \log_2 p \rceil$)	1024	2048	2048[1]
Bit length of subgroup order q (i.e., $\lceil \log_2 q \rceil$)	160	224	256

- No extra benefit from using small subgroups when already using short exponents
- DSA needs small subgroups, but not DH

Fast internet scanning lets us study behavior of publicly accessible hosts.

Widely deployed RFC5114 groups follow NIST recommendations*:

Group			Host Counts			
Name	**p (bits)**	**q (bits)**	**HTTPS**	**SMTP**	**IKEv1**	**IKEv2**
Group 22	1024	160	3%	$\approx$ 0%	17%	13%
Group 23	2048	224	$\approx$ 0%	33%	17%	13%
Group 24	2048	256	$\approx$ 0%	$\approx$ 0%	18%	14%
Total	—	—	40.6M	3.4M	1.9M	1.3M

Group 23: Can recover **201** bits of exponent in $\approx 2^{42}$ work

*: Scans from November 2016

Hosts don't validate group order.

	DHE Hosts	Non-Safe Primes	Hosts accepting...
HTTPS	11M	14%	
IKEv1	2.6M	13%	
IKEv2	1.3M	14%	
SSH	11M	$\approx$ 0%	

Hosts don't validate group order.

	DHE Hosts	Non-Safe Primes	Hosts accepting... 0
HTTPS	11M	14%	0.6%
IKEv1	2.6M	13%	*
IKEv2	1.3M	14%	*
SSH	11M	$\approx$ 0%	3%

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.

Hosts don't validate group order.

	DHE Hosts	Non-Safe Primes	Hosts accepting... 0	1
HTTPS	11M	14%	0.6%	3%
IKEv1	2.6M	13%	*	28%
IKEv2	1.3M	14%	*	0%
SSH	11M	$\approx$ 0%	3%	25%

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.

Hosts don't validate group order.

	DHE Hosts	**Non-Safe Primes**	**Hosts accepting...** 0	1	p-1
HTTPS	11M	14%	0.6%	3%	5%
IKEv1	2.6M	13%	*	28%	27%
IKEv2	1.3M	14%	*	0%	0%
SSH	11M	$\approx$ 0%	3%	25%	33%

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.

Hosts don't validate group order.

	DHE Hosts	**Non-Safe Primes**	**Hosts accepting...** 0	1	p-1	g_3/g_7
HTTPS	11M	14%	0.6%	3%	5%	$\approx 100\%$
IKEv1	2.6M	13%	*	28%	27%	99%
IKEv2	1.3M	14%	*	0%	0%	97%
SSH	11M	$\approx 0\%$	3%	25%	33%	N/A

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.

Libraries don't validate group order.

Similar findings to [DCE 2017 (up next!)]

Library (TLS)	**Validation**
Mozilla NSS	$g \leq 2$
OpenJDK	$g \leq 2$
OpenSSL 1.0.2	None*
BouncyCastle	$g \leq 2$
Cryptlib	$g \leq 2$
libTomCrypt	None
CryptoPP	None
Botan	None
GnuTLS	$g \leq 2$

*: before CVE-2016-0701 in Jan '16

- "The server obtains the DH parameters via a `PKCS#3` file which does not contain any subgroup information. This file format is the defacto standard across all crypto libraries."
- OpenSSL vulnerable to full Lim-Lee key recovery attack for RFC 5114 primes
- Amazon Load Balancer vulnerable to partial key recovery attack

Misconceptions

Academics

"There are many good reasons for using smaller subgroups, including **efficiency** and the fact that this setting matches the **theoretical security analyses of cryptosystems**."

Implementors

"safe primes (...) have quite some undesirable properties. They don't have a subgroup with size of the selected security parameter and that **requires them to use very large keys**."

Fact: Short exponents with safe primes and with small subgroups are both well-studied

Disconnects

Academics

"(...) it is only necessary to validate cryptographic parameters properly - **but this is very well-known**."

Implementors

"I bet there are TLS clients (and other DH users) out there that use those values, and we would break them (...) **functionality trumps security every day, and twice on Tuesdays**."

Countermeasures may be known, but are not always implemented

Takeaways

- Standards writers:
  - Software developers have different priorities
  - The fewer checks required, the better! (Murphy's Law)

Takeaways

- Standards writers:
  - Software developers have different priorities
  - The fewer checks required, the better! (Murphy's Law)
- Software developers:
  - Take care when it comes to cryptographic validation
  - Project Wycheproof: test crypto libraries against known attacks (`https://github.com/google/wycheproof`)

Takeaways

- Standards writers:
  - Software developers have different priorities
  - The fewer checks required, the better! (Murphy's Law)
- Software developers:
  - Take care when it comes to cryptographic validation
  - Project Wycheproof: test crypto libraries against known attacks (`https://github.com/google/wycheproof`)
- Sysadmins:
  - Test your servers with our tools! (`https://github.com/eniac/crypscan`)

Takeaways

- Standards writers:
  - Software developers have different priorities
  - The fewer checks required, the better! (Murphy's Law)
- Software developers:
  - Take care when it comes to cryptographic validation
  - Project Wycheproof: test crypto libraries against known attacks (`https://github.com/google/wycheproof`)
- Sysadmins:
  - Test your servers with our tools! (`https://github.com/eniac/crypscan`)

Questions?

References

Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice
David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, Paul Zimmermann. *CCS 2015.* `weakdh.org`

Indiscreet Logs: Persistent Diffie-Hellman Backdoors in TLS
Kristen Dorey, Nicholas Chang-Fong, Aleksander Essex. *NDSS 2017*

Measuring small subgroup attacks against Diffie-Hellman

Luke Valenta[*], David Adrian[†], Antonio Sanso[‡], Shaanan Cohney[*], Joshua Fried[*], Marcella Hastings[*], J. Alex Halderman[†], Nadia Heninger[*]

[*]University of Pennsylvania
[†]University of Michigan
[‡]Adobe

February 28, 2017