Sharing a Secret
in Plain Sight

Gregory Quenell

The Setting: Alice and Bob want to have a private conversation using email or texting.

Alice

Bob

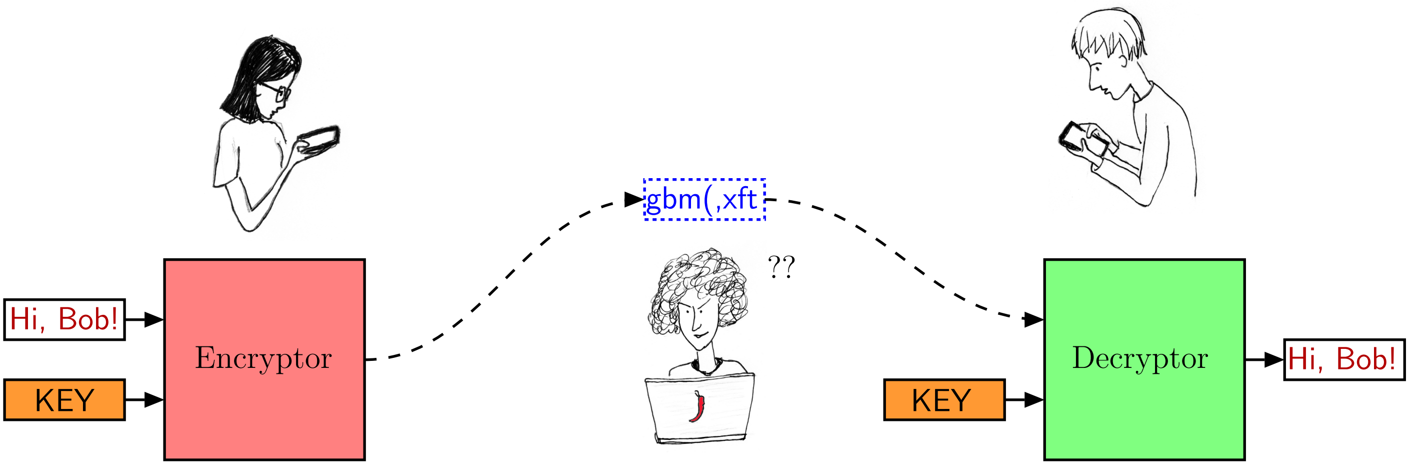
The Setting: Alice and Bob want to have a private conversation using email or texting.

The Problem: These media are insecure. Anyone can listen in by intercepting wifi packets.

The Solution: Encryption. In a *symmetric-key* or *shared-key* encryption scheme, Bob and Alice share a secret key (KEY) that Eve doesn't know.

Alice feeds KEY and her message into an Encryptor, which "locks" the message so that Eve can't read it.

Bob uses his copy of KEY in the Decryptor to "unlock" and read the message.



A New Problem: How can Alice and Bob agree on a shared key, while keeping it a secret from Eve?

A New Problem: How can Alice and Bob agree on a shared key, while keeping it a secret from Eve?

One possibility: Alice and Bob meet face-to-face, someplace where Eve can't hear them.

A New Problem: How can Alice and Bob agree on a shared key, while keeping it a secret from Eve?

One possibility: Alice and Bob meet face-to-face, someplace where Eve can't hear them.

But ...

They may not be able to do that. And anyway, if they could meet face-to-face, they could just have their private conversation then.

A New Problem: How can Alice and Bob agree on a shared key, while keeping it a secret from Eve?

One possibility: Alice and Bob meet face-to-face, someplace where Eve can't hear them.

But ...

They may not be able to do that. And anyway, if they could meet face-to-face, they could just have their private conversation then.

More realistically: Can Alice and Bob use the insecure channel, where Eve can intercept everything, to

- come up with a key that they both know, and
- keep it a secret from Eve?

A New Problem: How can Alice and Bob agree on a shared key, while keeping it a secret from Eve?

One possibility: Alice and Bob meet face-to-face, someplace where Eve can't hear them.

But . . .

They may not be able to do that. And anyway, if they could meet face-to-face, they could just have their private conversation then.

More realistically: Can Alice and Bob use the insecure channel, where Eve can intercept everything, to

- come up with a key that they both know, and
- keep it a secret from Eve?

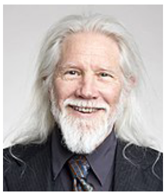
Surprisingly, the answer is yes.

The Diffie-Hellman Key Exchange Protocol

. . . allows two people who have no prior knowledge of one another to establish a

shared secret key

while communicating over an insecure channel.



The Diffie-Hellman protocol was first described in a 1976 paper by Whitfield Diffie and Martin Hellman.

Similar systems had previously been described by American cryptographer Ralph Merkle, and in classified research at the Government Communications Headquarters (GCHQ) in England.

Exponentiation in the ring $\mathbb{Z}_p$

For $g \in \mathbb{Z}_p$ and a positive integer k, we can compute $g^k \in \mathbb{Z}_p$ by repeatedly multiplying and reducing modulo p.

Example: 3^k in $\mathbb{Z}_7$.

$$
\begin{array}{rcccccccl}
 & & 3^1 & = & 3 & & & \equiv & 3 \pmod 7 \\
 & & 3^2 & = & 3 \times 3 & = & 9 & \equiv & 2 \pmod 7 \\
3^3 & = & 3 \times 3^2 & = & 3 \times 2 & = & 6 & \equiv & 6 \pmod 7 \\
3^4 & = & 3 \times 3^3 & = & 3 \times 6 & = & 18 & \equiv & 4 \pmod 7 \\
3^5 & = & 3 \times 3^4 & = & 3 \times 4 & = & 12 & \equiv & 5 \pmod 7 \\
3^6 & = & 3 \times 3^5 & = & 3 \times 5 & = & 15 & \equiv & 1 \pmod 7 \\
3^7 & = & 3 \times 3^6 & = & 3 \times 1 & = & 3 & \equiv & 3 \pmod 7 \\
\vdots & & & & \vdots & & & & \vdots
\end{array}
$$

The powers of 3 in the system $\mathbb{Z}_7$ look like this:

n	0	1	2	3	4	5	6	7	8	$\cdots$
3^n	1	3	2	6	4	5	1	3	2	$\cdots$

Exponentiation in the ring $\mathbb{Z}_p$

There's a much faster method, called the square-and-multiply method.

Example: 2^{68} in $\mathbb{Z}_{101}$.

$$
\begin{array}{lllllllll}
2^2 & & & & & = & 4 & \equiv & 4 \pmod{101} \\
2^4 & = & (2^2)^2 & = & 4^2 & = & 16 & \equiv & 16 \pmod{101} \\
2^8 & = & (2^4)^2 & = & 16^2 & = & 256 & \equiv & 54 \pmod{101} \\
2^{16} & = & (2^8)^2 & = & 54^2 & = & 2916 & \equiv & 88 \pmod{101} \\
2^{32} & = & (2^{16})^2 & = & 88^2 & = & 7744 & \equiv & 68 \pmod{101} \\
2^{64} & = & (2^{32})^2 & = & 68^2 & = & 4624 & \equiv & 79 \pmod{101}
\end{array}
$$

Exponentiation in the ring $\mathbb{Z}_p$

There's a much faster method, called the square-and-multiply method.

Example: 2^{68} in $\mathbb{Z}_{101}$.

$$
\begin{array}{rrcrcrcrcl}
 & 2^2 & & & & & = & 4 & \equiv & 4 \pmod{101} \\
\rightarrow & 2^4 & = & (2^2)^2 & = & 4^2 & = & 16 & \equiv & 16 \pmod{101} \\
 & 2^8 & = & (2^4)^2 & = & 16^2 & = & 256 & \equiv & 54 \pmod{101} \\
 & 2^{16} & = & (2^8)^2 & = & 54^2 & = & 2916 & \equiv & 88 \pmod{101} \\
 & 2^{32} & = & (2^{16})^2 & = & 88^2 & = & 7744 & \equiv & 68 \pmod{101} \\
\rightarrow & 2^{64} & = & (2^{32})^2 & = & 68^2 & = & 4624 & \equiv & 79 \pmod{101}
\end{array}
$$

Now $2^{68} = 2^{64+4} = 2^{64} \times 2^4$, so we get

$$2^{68} = (2^{64}) \times (2^4) = 79 \times 16 = 1264 \equiv 52 \pmod{101}.$$

Exponentiation in the ring $\mathbb{Z}_p$

There's a much faster method, called the square-and-multiply method.

Example: 2^{68} in $\mathbb{Z}_{101}$.

$$\begin{array}{rrcrcrcrcl} & 2^2 & & & & & = & 4 & \equiv & 4 \pmod{101} \\ \rightarrow & 2^4 & = & (2^2)^2 & = & 4^2 & = & 16 & \equiv & 16 \pmod{101} \\ & 2^8 & = & (2^4)^2 & = & 16^2 & = & 256 & \equiv & 54 \pmod{101} \\ & 2^{16} & = & (2^8)^2 & = & 54^2 & = & 2916 & \equiv & 88 \pmod{101} \\ & 2^{32} & = & (2^{16})^2 & = & 88^2 & = & 7744 & \equiv & 68 \pmod{101} \\ \rightarrow & 2^{64} & = & (2^{32})^2 & = & 68^2 & = & 4624 & \equiv & 79 \pmod{101} \end{array}$$

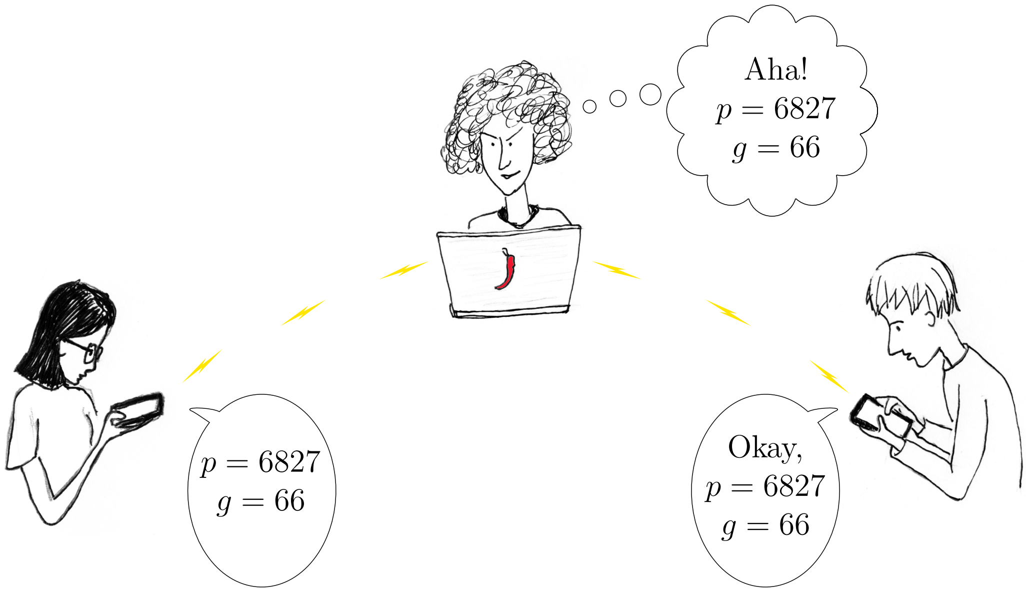
Now $2^{68} = 2^{64+4} = 2^{64} \times 2^4$, so we get

$$2^{68} = (2^{64}) \times (2^4) = 79 \times 16 = 1264 \equiv 52 \pmod{101}.$$

Cost: seven multiplications (with reductions modulo 101).

Alice and Bob

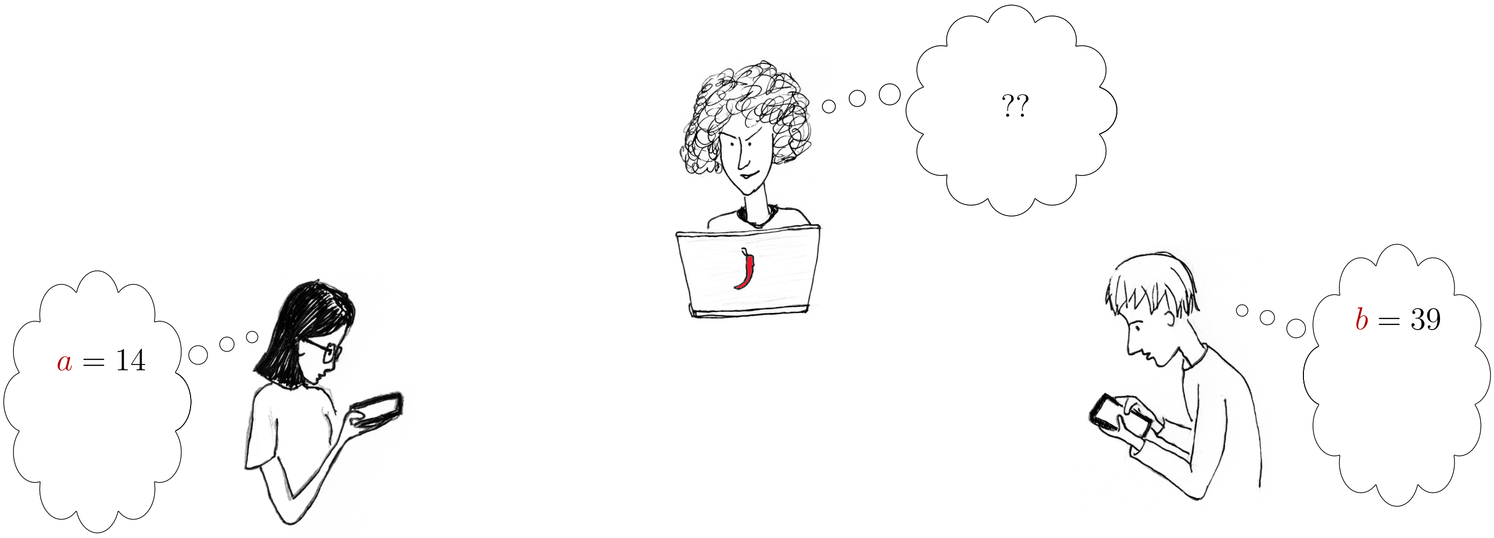
- Alice selects a prime p and a base g, and sends these to Bob.
- Eve listens in; she now knows the values of p and g.



Alice and Bob

$(p = 6827; g = 66)$

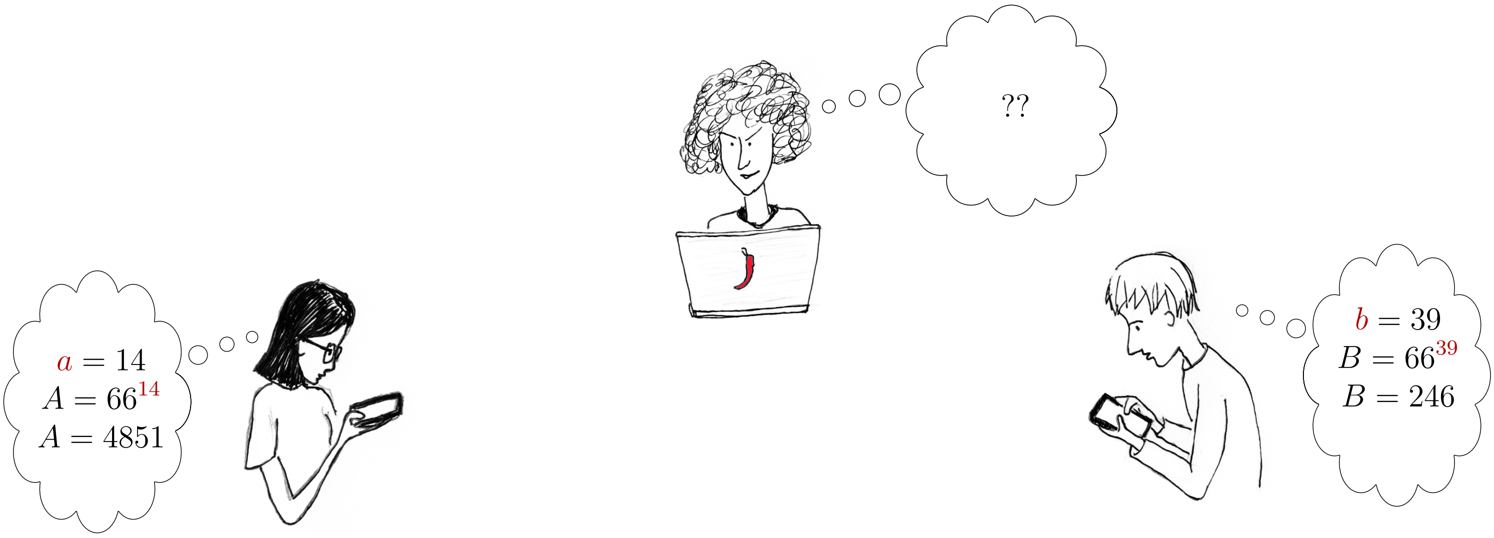
- Alice chooses a secret exponent a, and doesn't tell anyone.
- Bob chooses a secret exponent b, and doesn't tell anyone.



Alice and Bob

$(p = 6827;\ g = 66)$

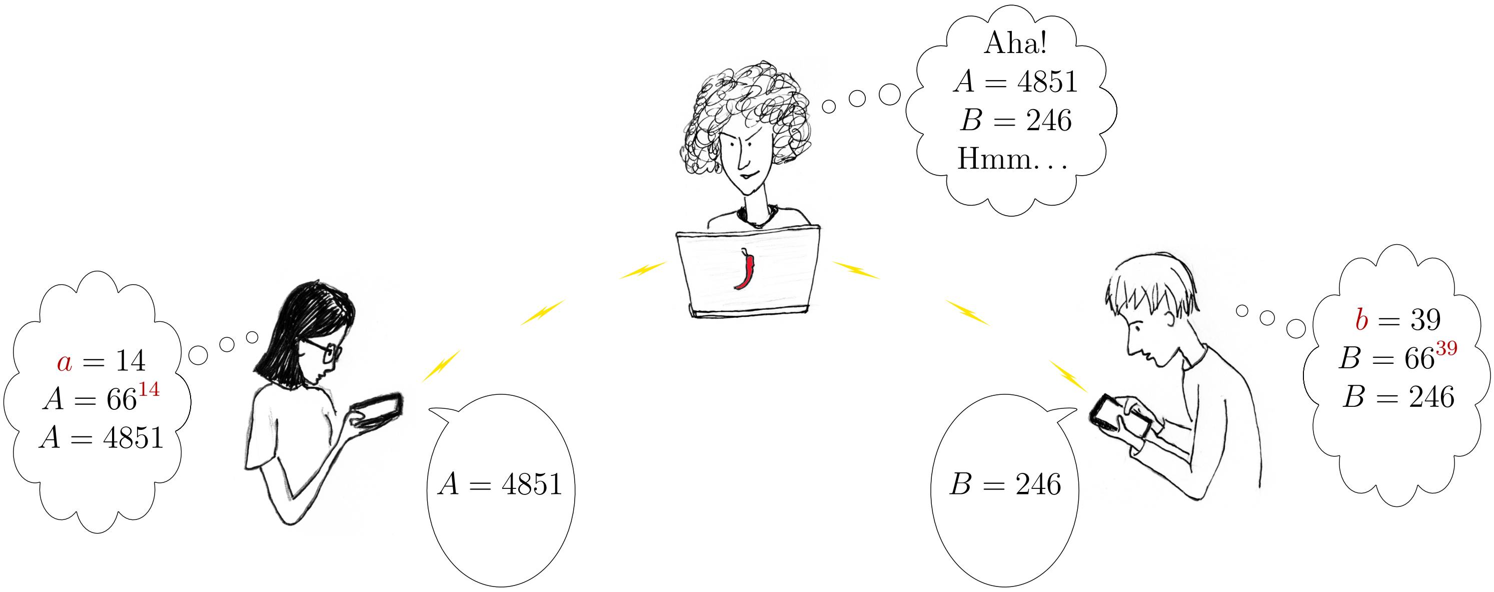
- Alice uses square-and-multiply to compute $A = g^{a}$.
- Bob uses square-and-multiply to compute $B = g^{b}$.



Alice and Bob

$(p = 6827;\ g = 66)$

- Alice sends A to Bob, and Bob sends B to Alice.
- Eve now knows the values of g, $A = g^a$, and $B = g^b$.

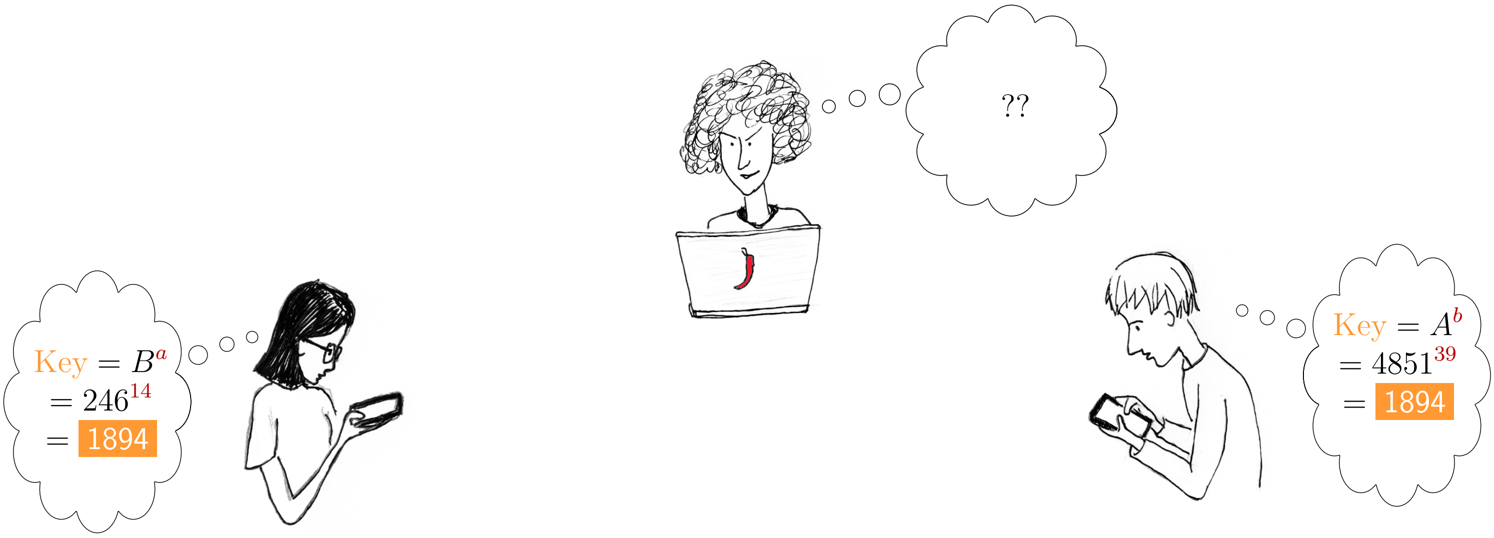


Alice and Bob

$(p = 6827;\ g = 66)$

- Bob now computes A^b, which is equal to $(g^a)^b$.
- Alice now computes B^a, which is equal to $(g^b)^a$.

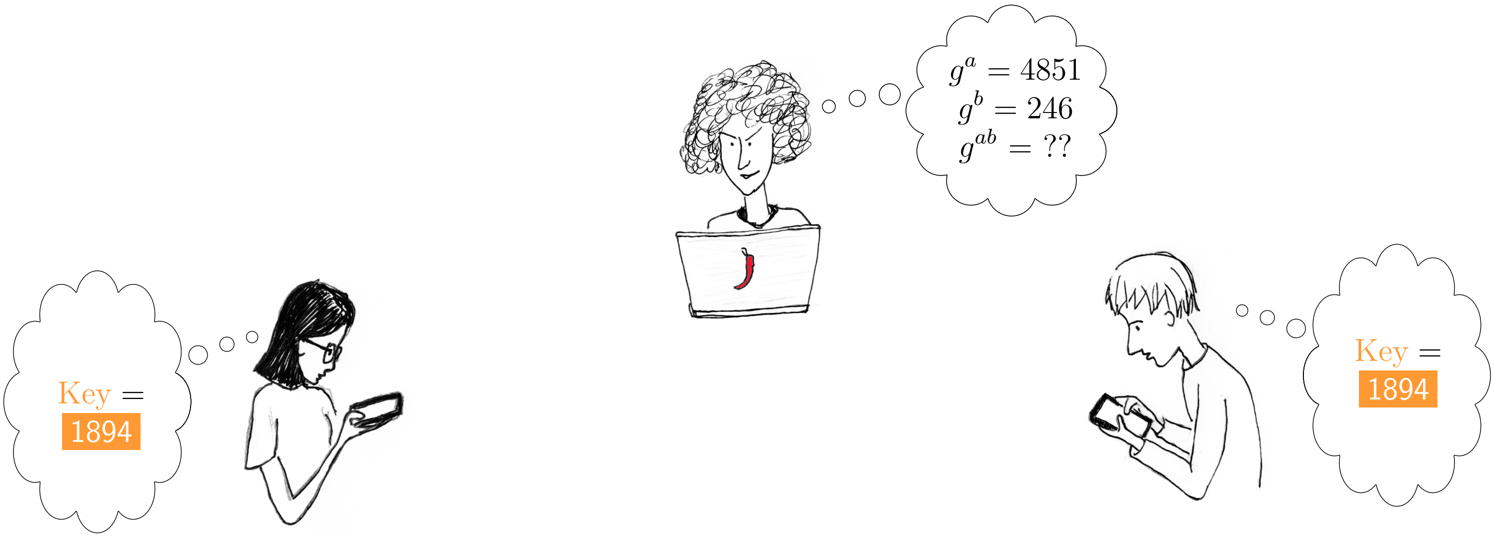
By the laws of exponents, $(g^a)^b = g^{ab} = (g^b)^a$, so Bob and Alice have the *same number*.



Alice and Bob

$(p = 6827; g = 66)$

- Alice and Bob, each using square-and-multiply twice, have independently calculated g^{ab}.
- Eve, by listening in, has learned g, g^a and g^b, but has no good way to calculate g^{ab} from this information.



Security

The secrecy of the shared key g^{ab} relies on the

Computational Diffie-Hellman Assumption

CDH: Let g be a generator of a cyclic group G. Given generic elements g^a and g^b, Eve has no efficient algorithm for finding g^{ab}:

$$(g^a, g^b) \not\longmapsto g^{ab}$$

Security

The secrecy of the shared key g^{ab} relies on the

Computational Diffie-Hellman Assumption

CDH: Let g be a generator of a cyclic group G. Given generic elements g^a and g^b, Eve has no efficient algorithm for finding g^{ab}:

$$(g^a, g^b) \not\mapsto g^{ab}$$

Proof: There is none. It's an assumption. (And it depends on G.)

Security

The secrecy of the shared key g^{ab} relies on the

<u>*Computational Diffie-Hellman Assumption*</u>

> CDH: Let g be a generator of a cyclic group G. Given generic elements g^a and g^b, Eve has no efficient algorithm for finding g^{ab}:
>
> $$(g^a, g^b) \not\mapsto g^{ab}$$

Proof: There is none. It's an assumption. (And it depends on G.)

Belief: For certain cyclic groups, Eve's best approach is to find either a or b, the *discrete logarithms* of g^a and g^b.

> DLP (the Discrete Logarithm Problem): Let g be a generator of a cyclic group G. Given a generic element g^a, find a.

Security

Further belief: For certain cryptographic groups, the fastest DLP algorithms require somewhat more than $\sqrt{|G|}$ arithmetic operations.

Security

Further belief: For certain cryptographic groups, the fastest DLP algorithms require somewhat more than $\sqrt{|G|}$ arithmetic operations.

Actual fact: The square-and-multiply algorithm for finding $(g^a)^b$ or $(g^b)^a$ requires at most $2 \log_2 |G|$ operations.

Security

Further belief: For certain cryptographic groups, the fastest DLP algorithms require somewhat more than $\sqrt{|G|}$ arithmetic operations.

Actual fact: The square-and-multiply algorithm for finding $(g^a)^b$ or $(g^b)^a$ requires at most $2\log_2|G|$ operations.

Result: In our toy example, $|G| = 6826$, so Bob and Alice can find their shared key g^{ab} using no more than

$$2\log_2 6826 \approx 26 \text{ multiplications.}$$

In order to find a and "break in", Eve's best algorithm would require at least

$$\sqrt{6826} \approx 83 \text{ multiplications.}$$

Security

Further belief: For certain cryptographic groups, the fastest DLP algorithms require somewhat more than $\sqrt{|G|}$ arithmetic operations.

Actual fact: The square-and-multiply algorithm for finding $(g^a)^b$ or $(g^b)^a$ requires at most $2\log_2|G|$ operations.

Result: In our toy example, $|G| = 6826$, so Bob and Alice can find their shared key g^{ab} using no more than

$$2\log_2 6826 \approx 26 \text{ multiplications.}$$

In order to find a and “break in”, Eve's best algorithm would require at least

$$\sqrt{6826} \approx 83 \text{ multiplications.}$$

In the toy example, Eve's task is trivial. There is no secrecy here. However . . .

Security

$\|G\|$	$\sqrt{\|G\|}$	$\log_2 \|G\|$	Time at 10^9/second Alice/Bob	Eve
10^5	300	17	4.6 μs	87 μs
10^{15}	3×10^7	33	120 μs	78 sec
10^{25}	3×10^{12}	83	570 μs	250 days
10^{35}	3×10^{17}	116	1.6 ms	1.4×10^5 yr
10^{45}	3×10^{22}	149	3.3 ms	2.2×10^{10} yr
10^{50}	10^{25}	166	4.6 ms	8.7×10^{12} yr

Security

			Time at 10^9/second	
$\lvert G\rvert$	$\sqrt{\lvert G\rvert}$	$\log_2 \lvert G\rvert$	Alice/Bob	Eve
10^5	300	17	4.6 μs	87 μs
10^{15}	3×10^7	33	120 μs	78 sec
10^{25}	3×10^{12}	83	570 μs	250 days
10^{35}	3×10^{17}	116	1.6 ms	1.4×10^5 yr
10^{45}	3×10^{22}	149	3.3 ms	2.2×10^{10} yr
10^{50}	10^{25}	166	4.6 ms	8.7×10^{12} yr

Moral: When $\lvert G\rvert \approx 10^{50}$, we get pretty good security.

Security

$\lvert G\rvert$	$\sqrt{\lvert G\rvert}$	$\log_2 \lvert G\rvert$	Time at 10^9/second Alice/Bob	Eve
10^5	300	17	4.6 μs	87 μs
10^{15}	3×10^7	33	120 μs	78 sec
10^{25}	3×10^{12}	83	570 μs	250 days
10^{35}	3×10^{17}	116	1.6 ms	1.4×10^5 yr
10^{45}	3×10^{22}	149	3.3 ms	2.2×10^{10} yr
10^{50}	10^{25}	166	4.6 ms	8.7×10^{12} yr

Moral: When $\lvert G\rvert \approx 10^{50}$, we get pretty good security.

(NIST recommends that $\lvert G\rvert$ should be at least 10^{68}, *and* that G should be hidden inside a bigger group with order at least 10^{616}.)

References

Dan Boneh and Victor Shoup, *A Graduate Course in Applied Cryptography*, prepublication version 0.4, 2017.

Whitfield Diffie and Martin E. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory, IT-22, November 1976.

Jonathan Katz and Yehuda Lindell, *Introduction to Modern Cryptography*, second edition, CRC Press, 2015.

Neal Koblitz, *A Course in Number Theory and Cryptography*, second edition, Springer-Verlag, 2006.

Douglas R. Stinson, *Cryptography: Theory and Practice*, second edition, CRC Press, 2002.