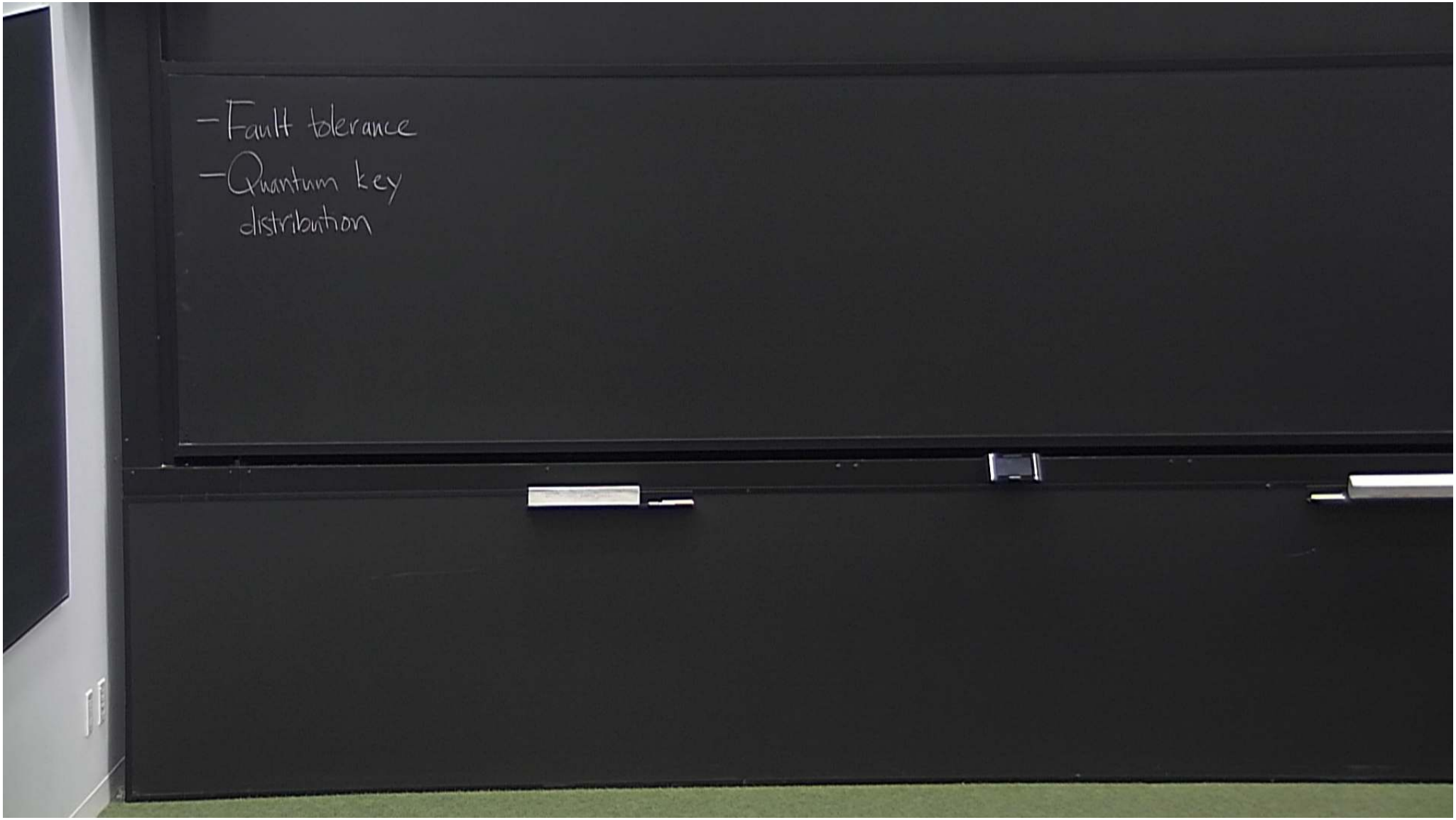
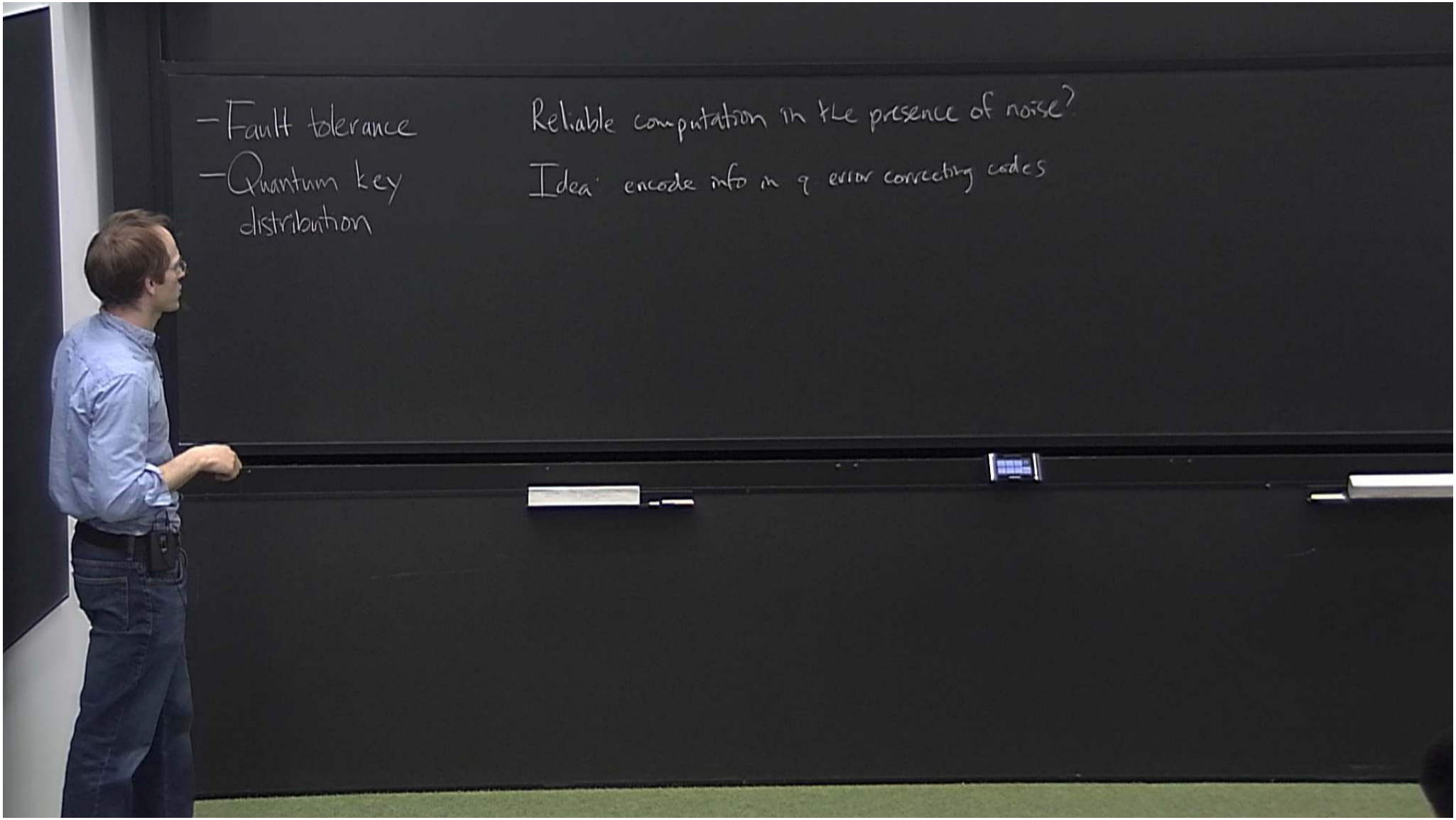
Title: 12/13 PSI - Quantum Information Review Lecture 14

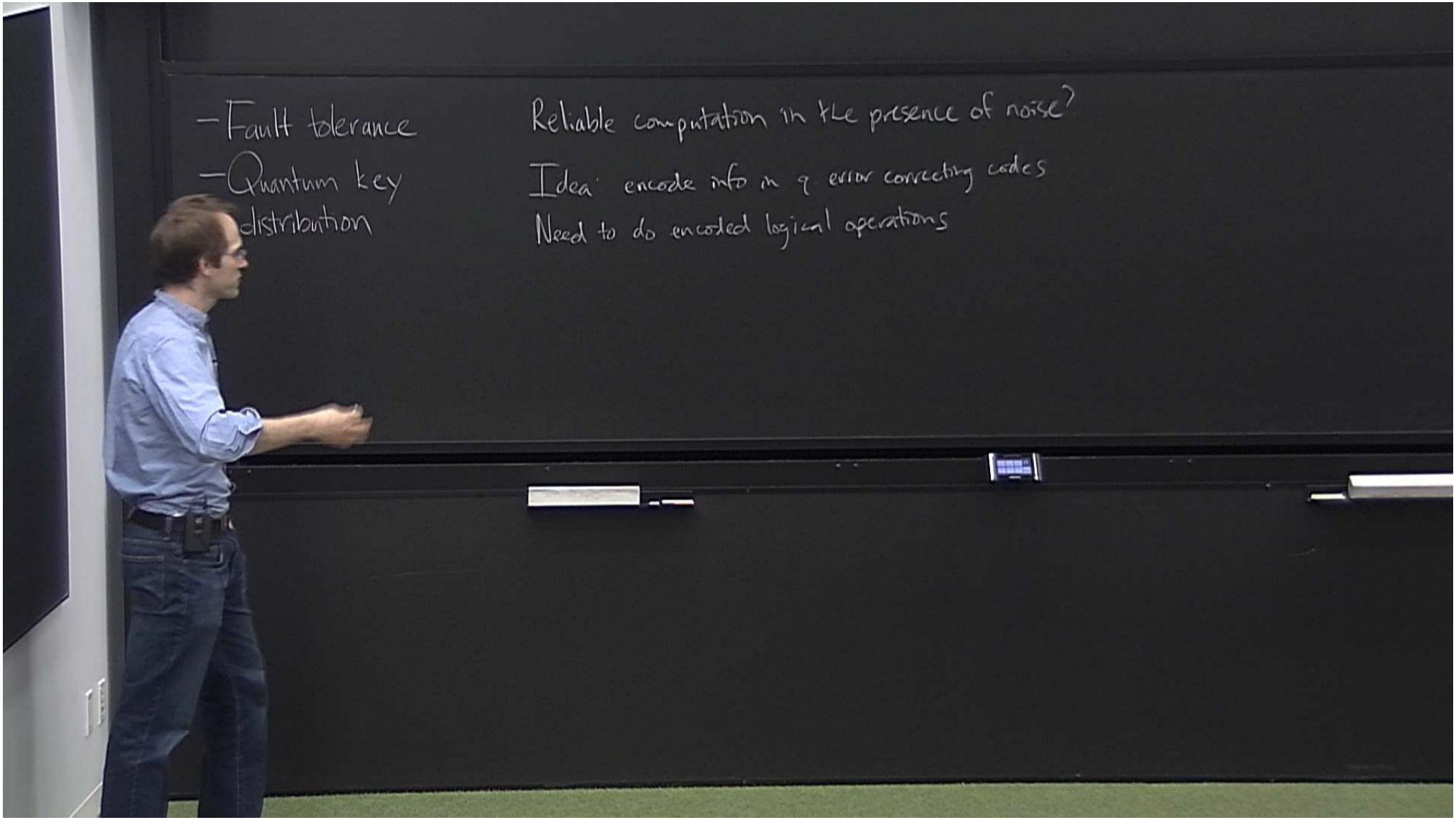
Date: Mar 08, 2013 10:15 AM

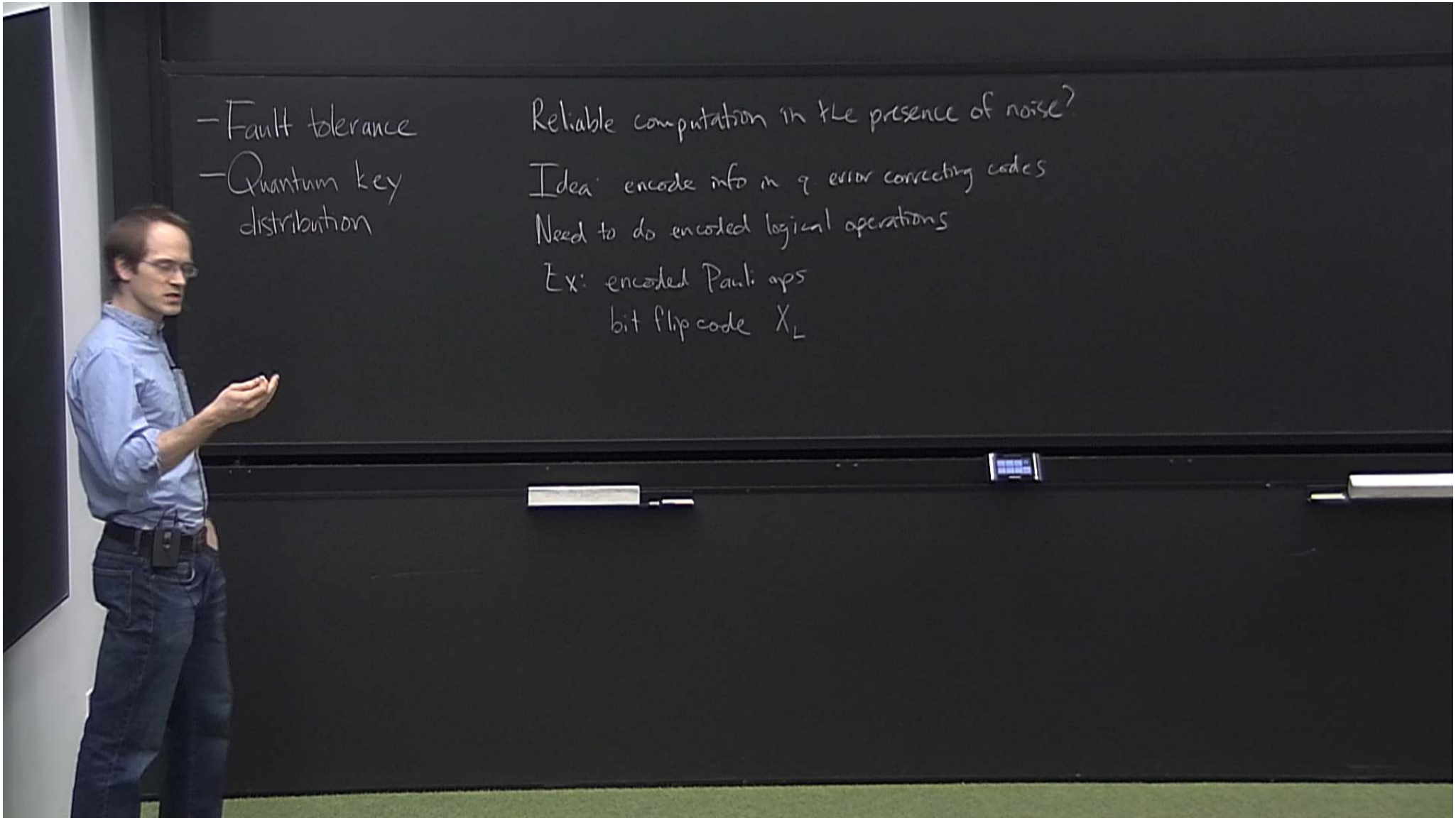
URL: http://www.pirsa.org/13030053

Abstract:


- Fault tolerance
- Quantum key distribution


- Fault tolerance
Reliable computation in the presence of noise?
- Quantum key distribution
Idea: encode info in q error correcting codes


-Fault tolerance
-Quantum key
distribution
Reliable computation in the presence of noise?
Idea: encode info in q error correcting codes
Need to do encoded logical operations


- Fault tolerance
- Quantum key distribution
Reliable computation in the presence of noise?
Idea: encode info in q error correcting codes
Need to do encoded logical operations
Ex: encoded Pauli ops
bit flip code X_L

Reliable computation in the presence of noise?

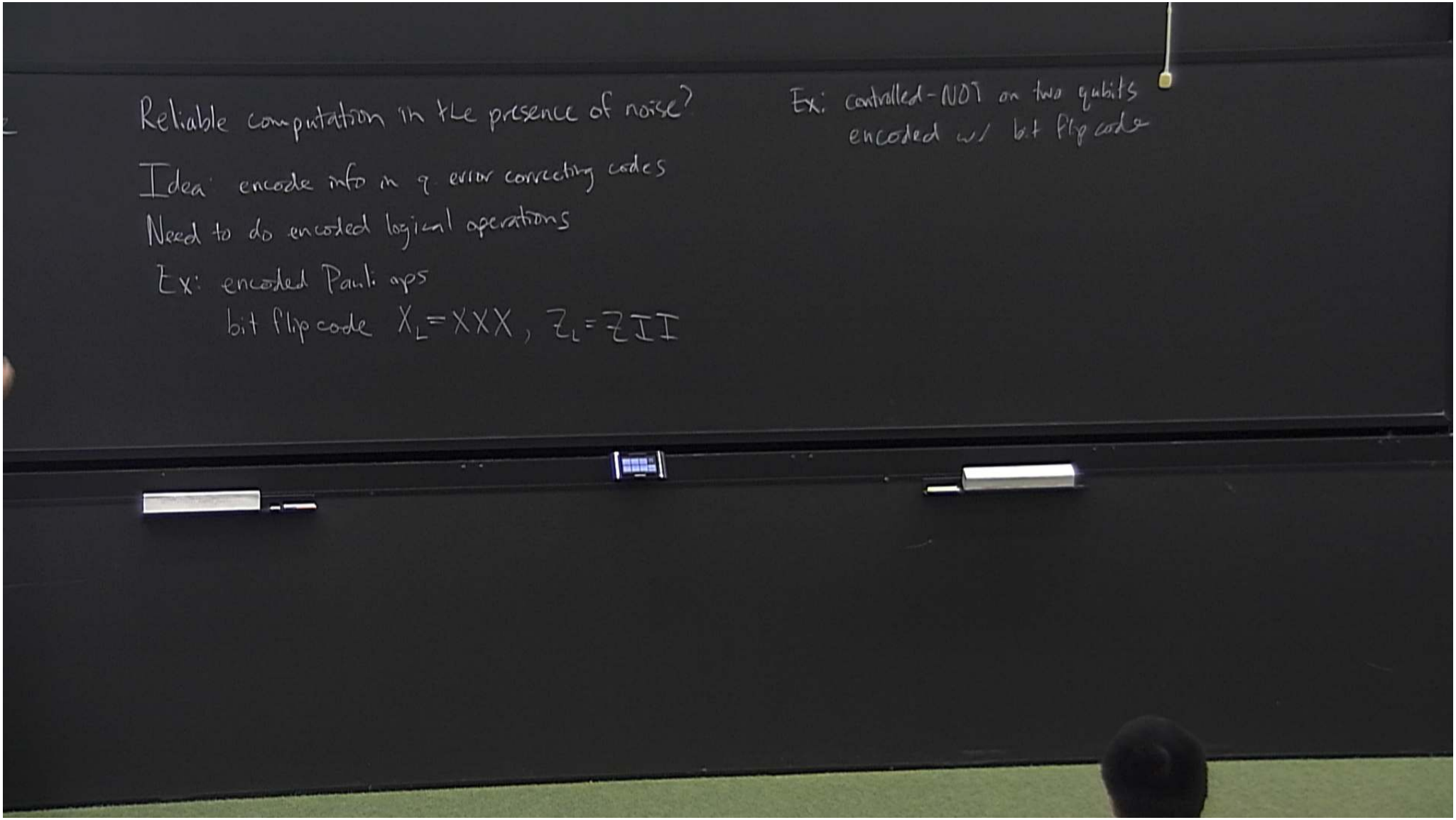
Idea: encode info in q. error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

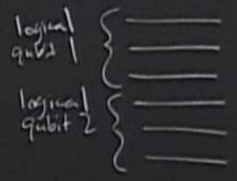
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

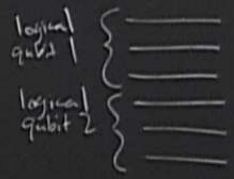
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



- Fault tolerance
- Quantum key distribution
Reliable computation in the presence of noise?
Idea: encode info in q error correcting codes
Need to do encoded logical operations
Ex: encoded Pauli ops
bit flip code $X_L = XXX$, $Z_L = ZII$
Ex: controlled-NOT on two qubits encoded w/ bit flip code
logical qubit 1
logical qubit 2

Reliable computation in the presence of noise?

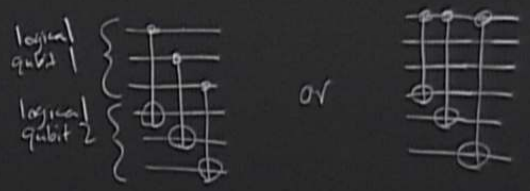
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

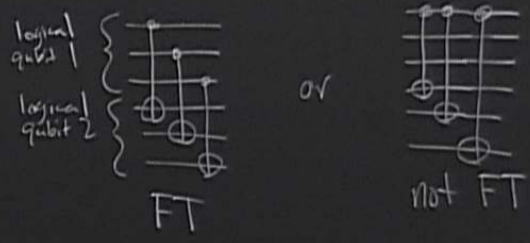
Idea: encode info in q error correcting codes

Need to d[illegible]ed logical operations

Ex: en[illegible]Pauli ops

[illegible]code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

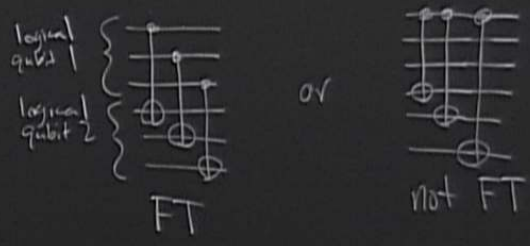
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

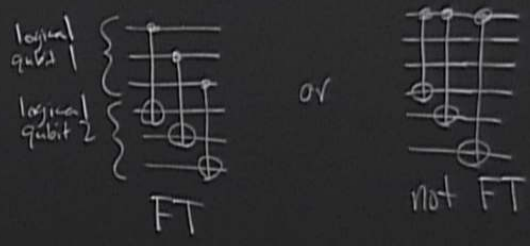
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

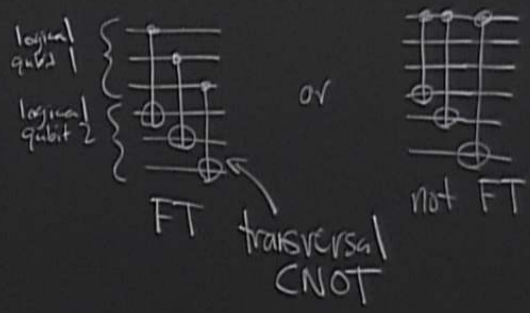
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Reliable computation in the presence of noise?

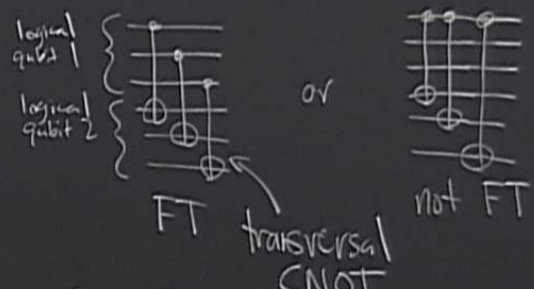
Idea: encode info in q error correcting codes

Need to do encoded logical operations

Ex: encoded Pauli ops

bit flip code $X_L = XXX$, $Z_L = ZII$

Ex: controlled-NOT on two qubits encoded w/ bit flip code



Steane 7-qubit lets us do lots of gates transversally

First, design fault-tolerant implementations of gates so that errors with prob p

$\rightarrow$ logical error rate $O(p^2)$.

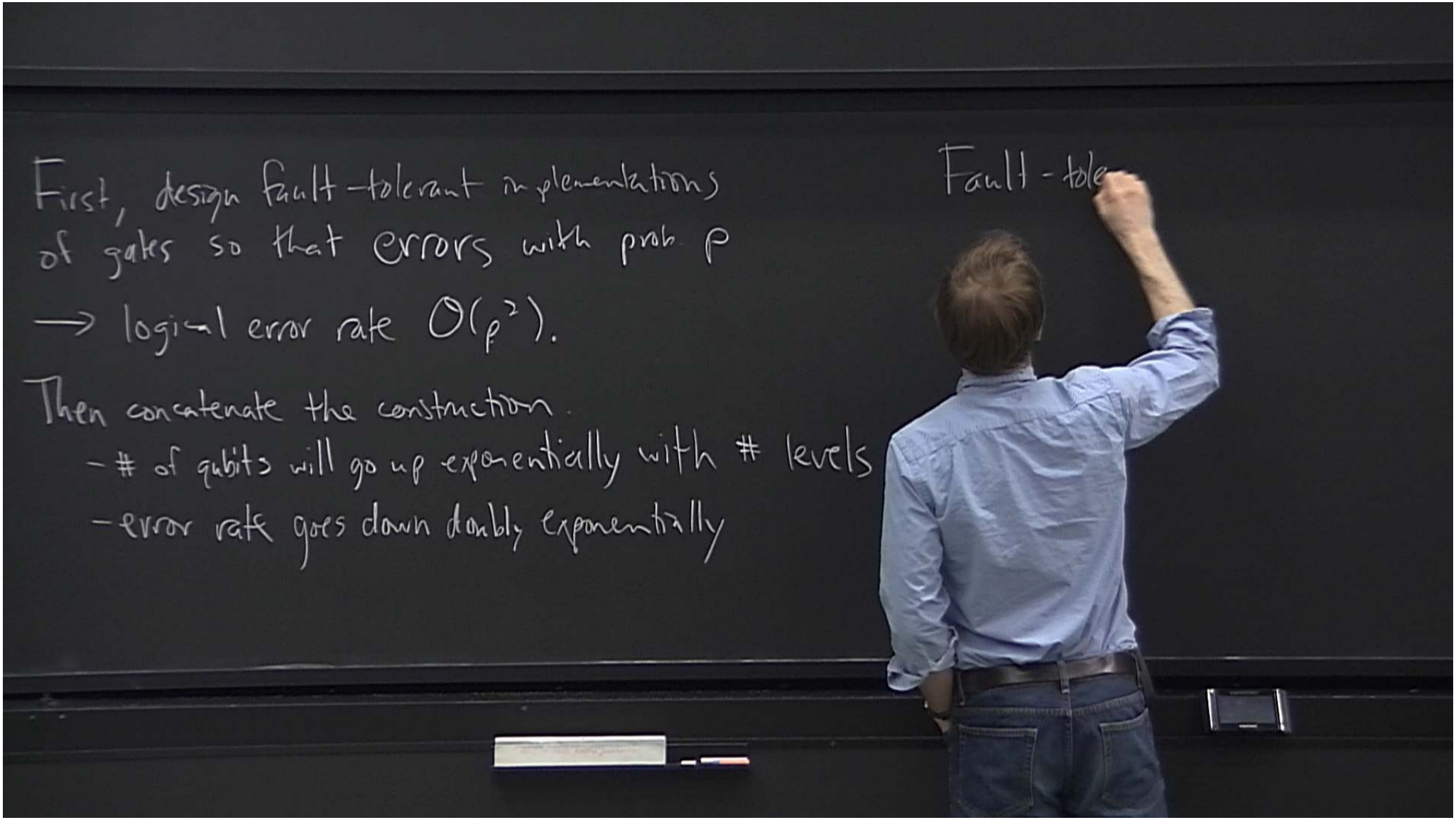
Then concatenate the construction.

First, design fault-tolerant implementations of gates so that errors with prob p

$\rightarrow$ logical error rate $O(p^2)$.

Then concatenate the construction.

- # of qubits will go up exponentially with # levels


First, design fault-tolerant implementations
of gates so that errors with prob p
→ logical error rate $O(p^2)$.
Then concatenate the construction.
- # of qubits will go up exponentially with # levels
- error rate goes down doubly exponentially
Fault-tole

...plementations

... with prob. p

...).

...on.

...entially with # levels

...y exponentially

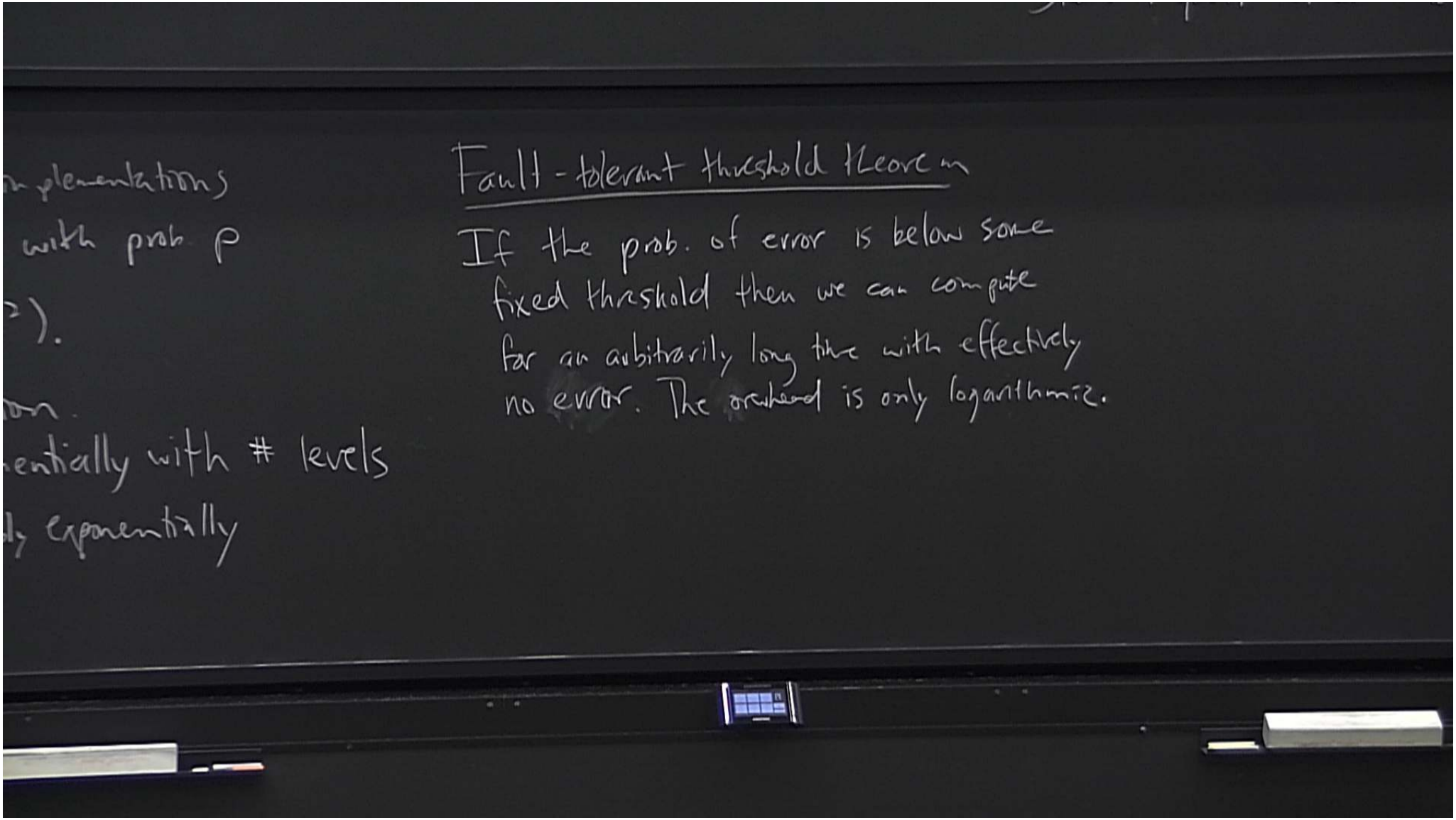
Fault-tolerant threshold theorem

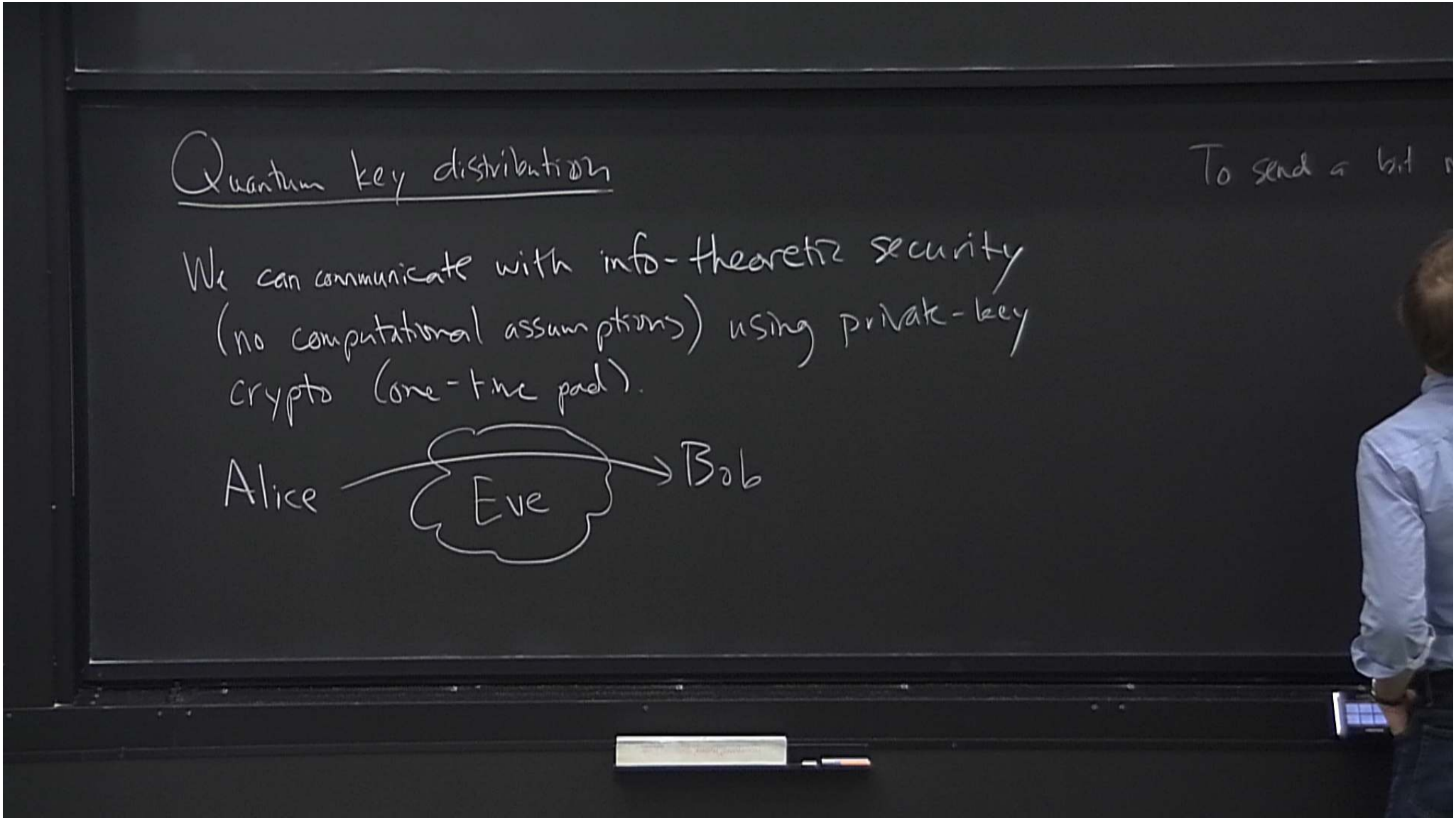
If the prob. of error is below some fixed threshold then we can compute for an arbitrarily long time with effectively no error.

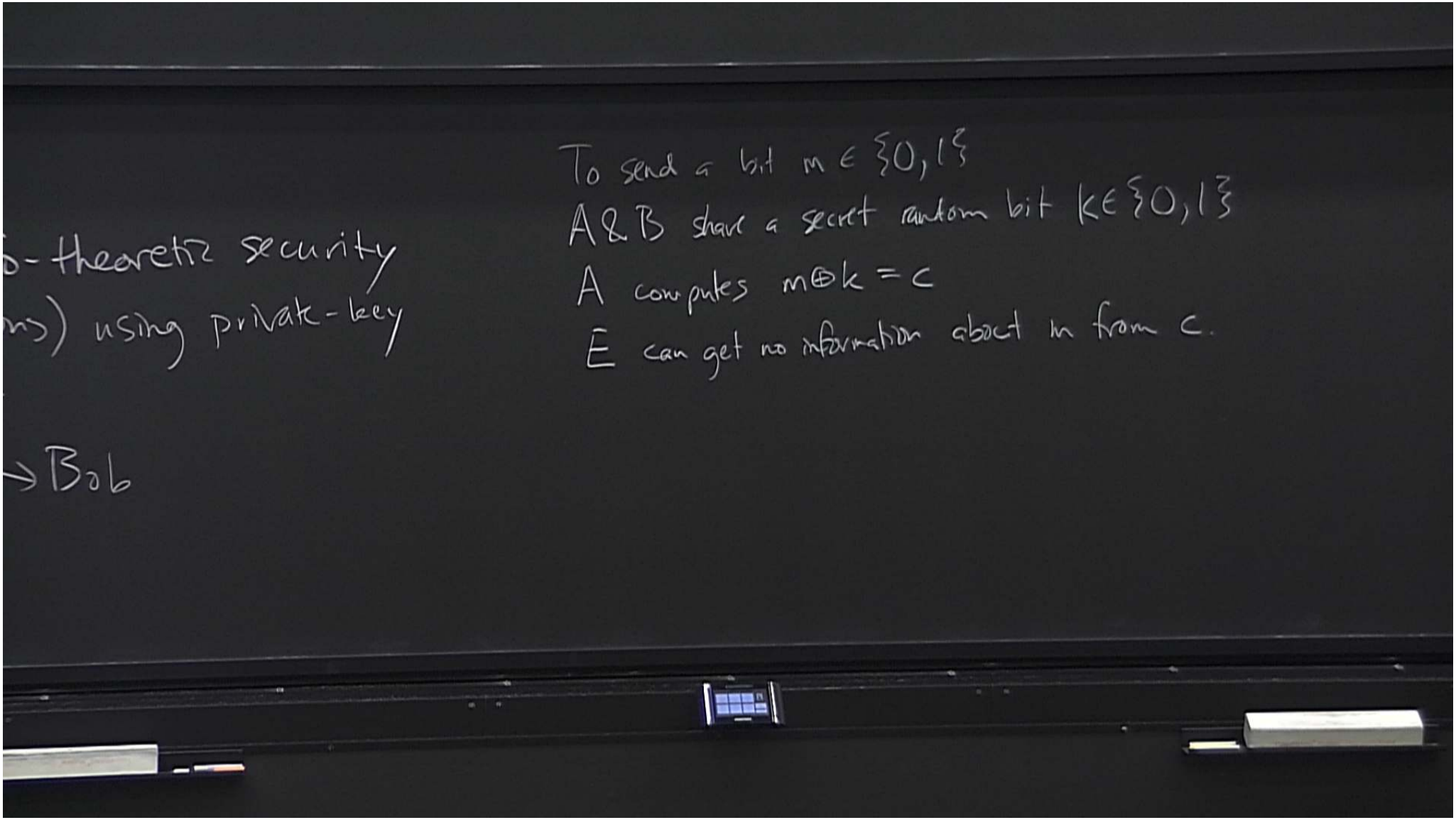
...plementations
... with prob p
...).
...ion.
...entially with # levels
...ly exponentially

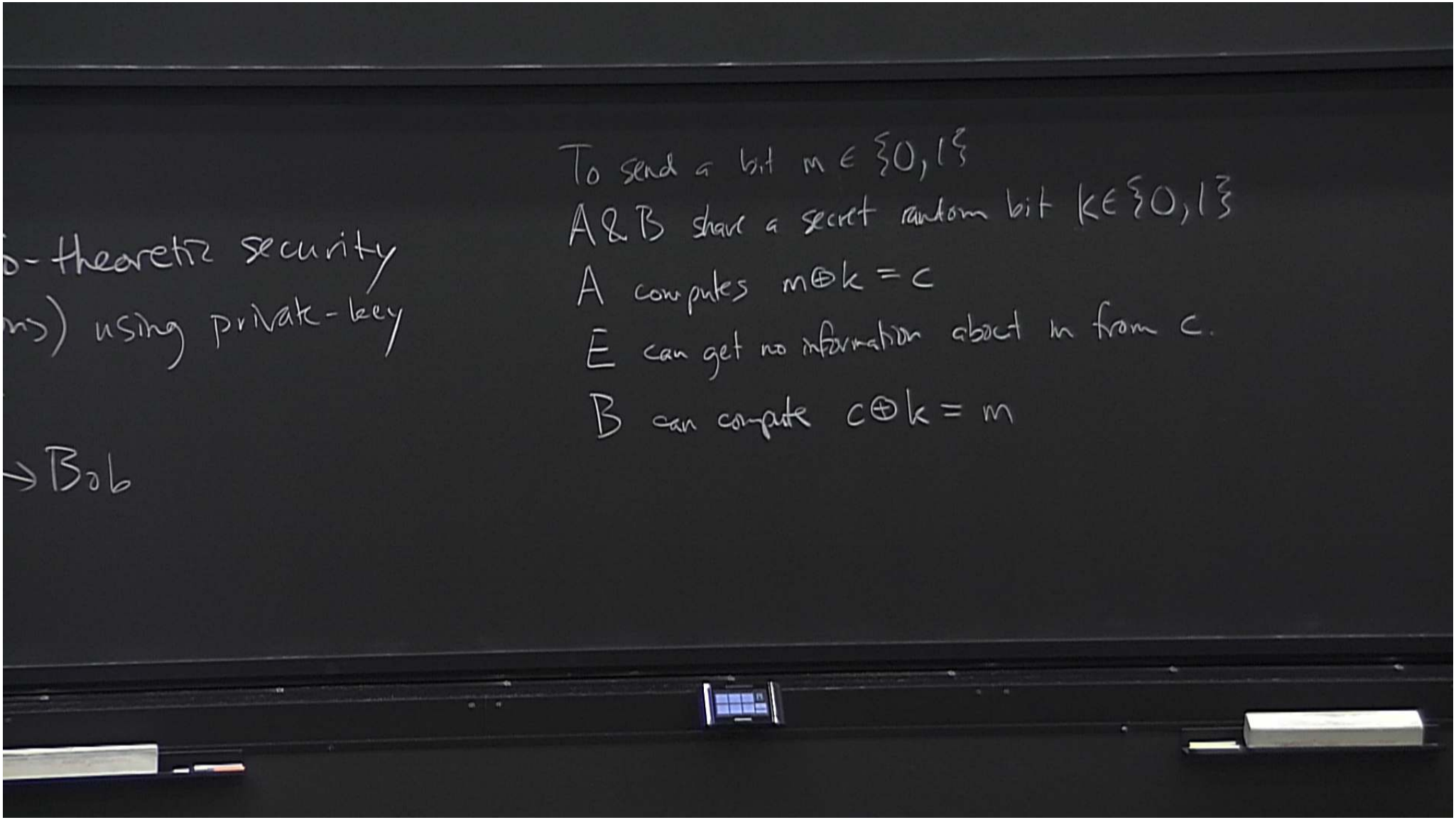
Fault-tolerant threshold theorem

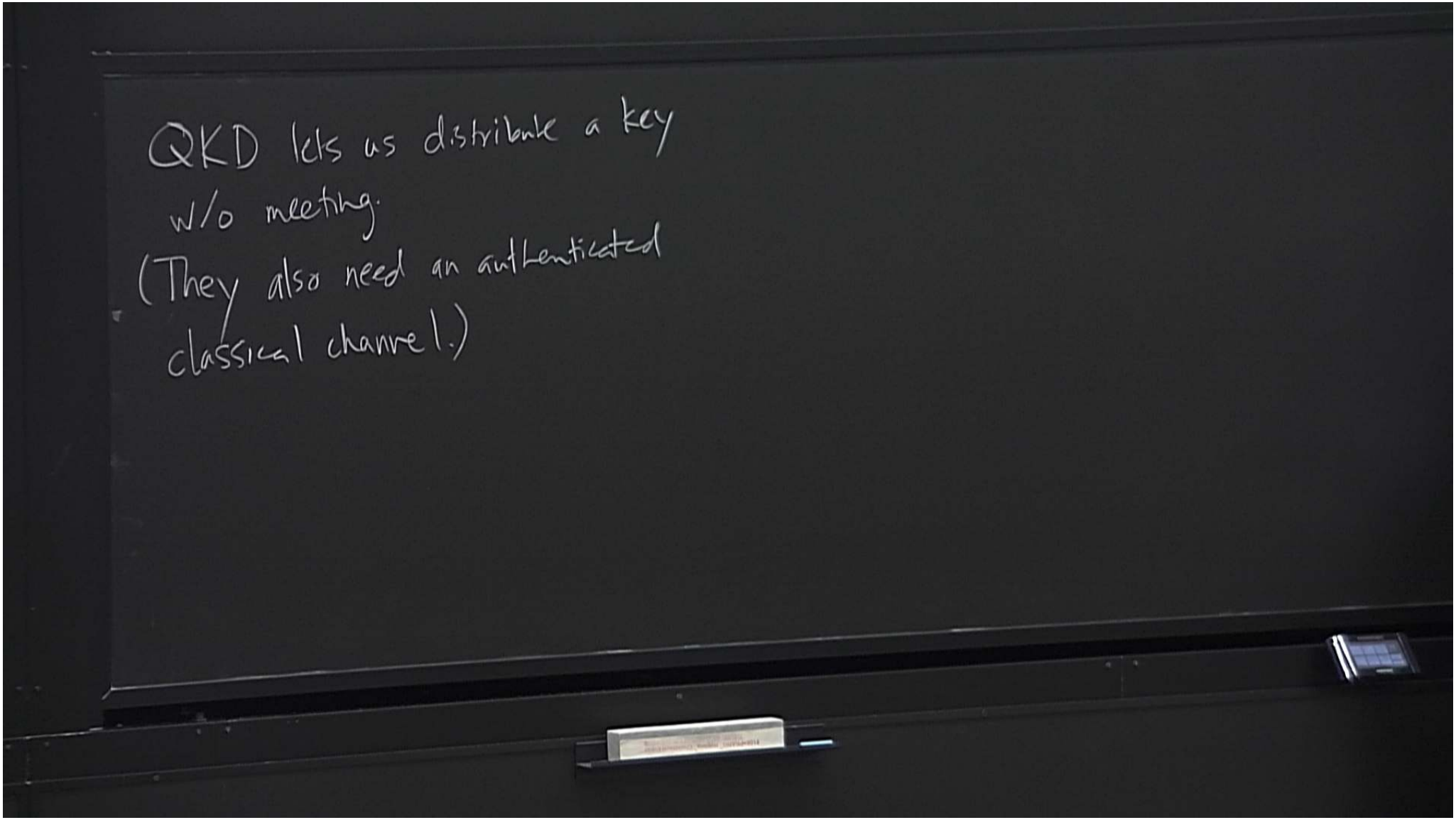
If the prob. of error is below some fixed threshold then we can compute for an arbitrarily long time with effectively no error. The overhead is only logarithmic.

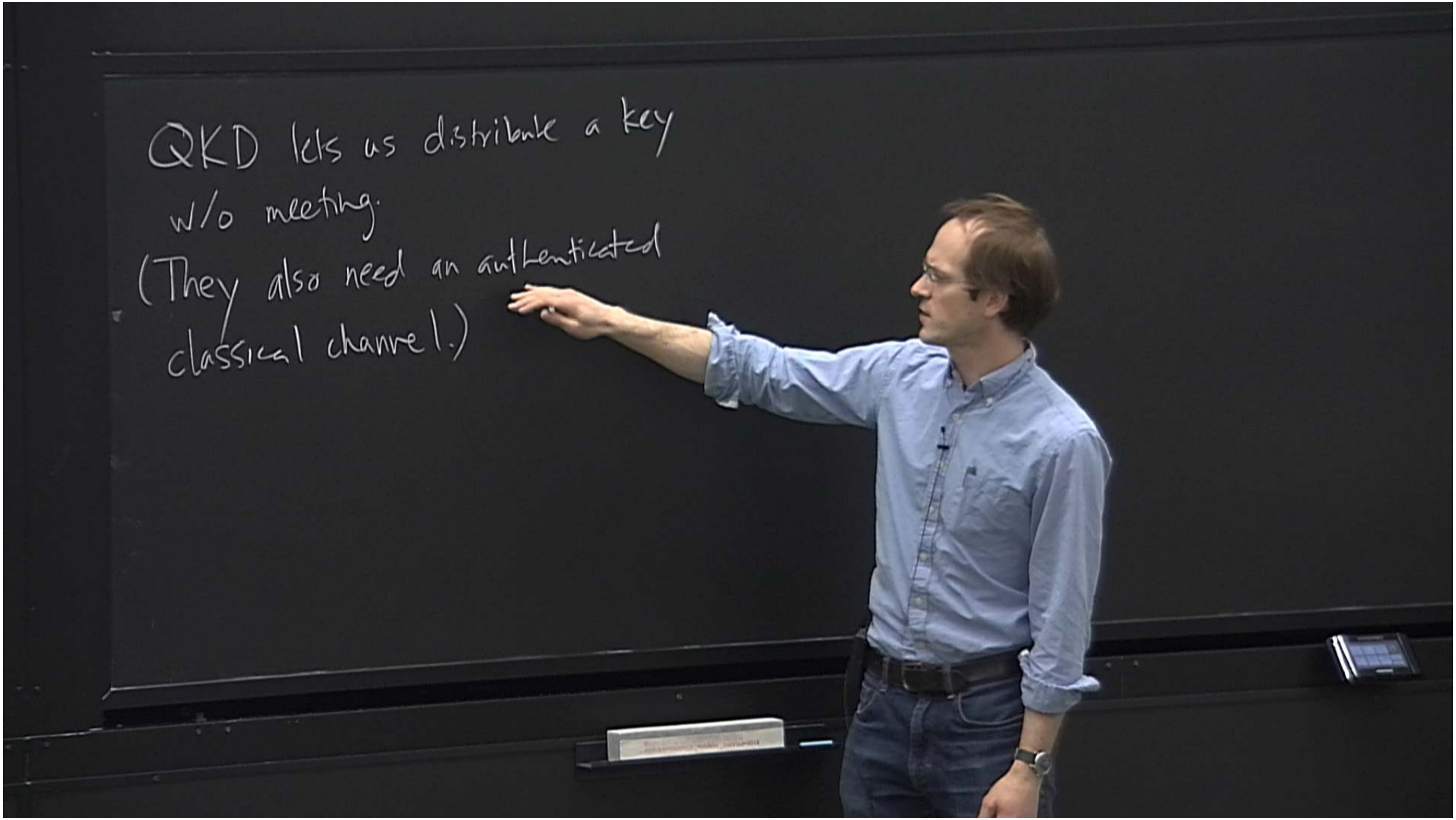

plementations)
with prob p
).
ion.
entially with # levels
ly exponentially
Fault-tolerant threshold theorem
If the prob. of error is below some
fixed threshold then we can compute
for an arbitrarily long time with effectively
no error. The overhead is only logarithmic.

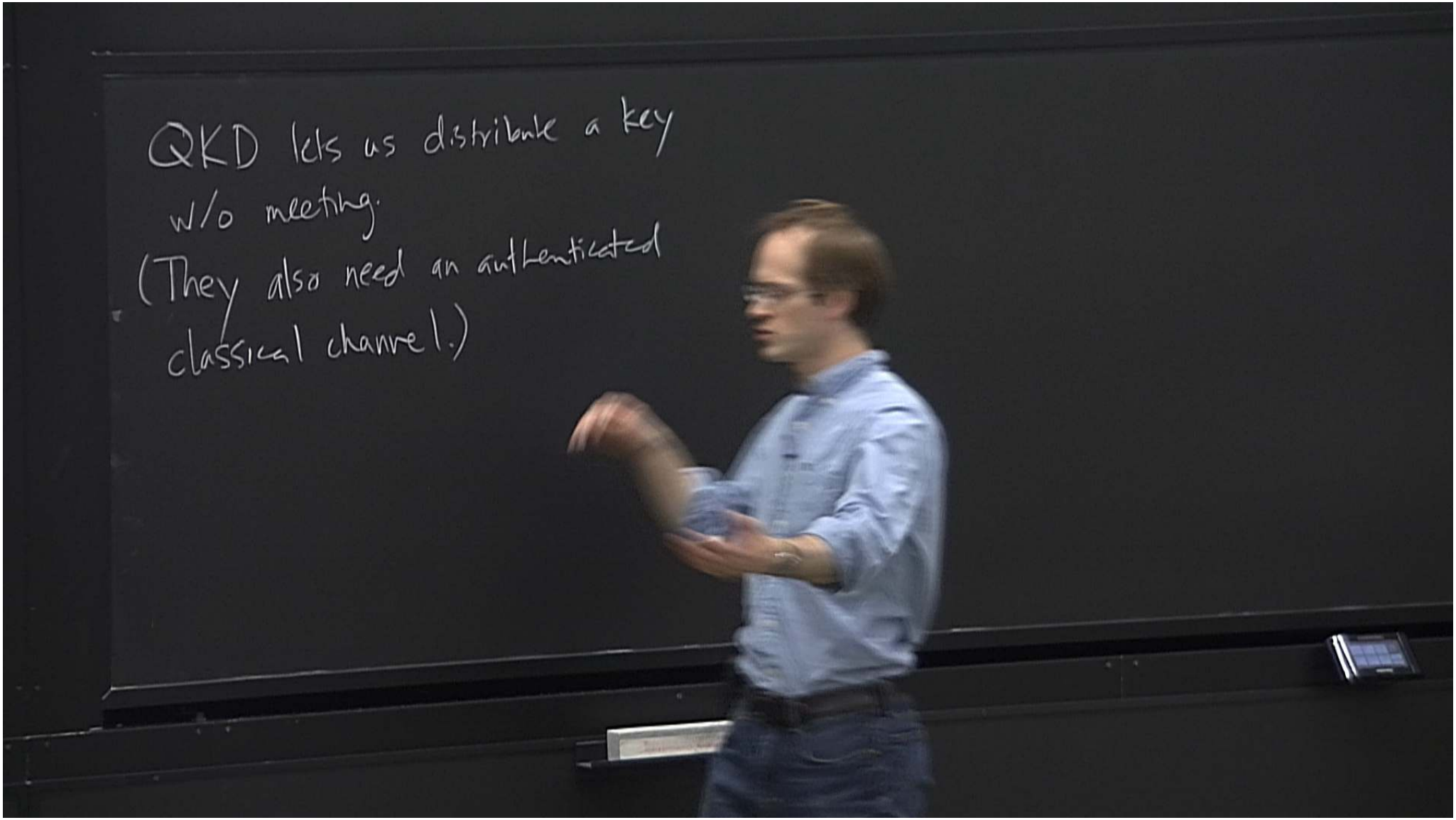

Quantum key distribution
To send a bit
We can communicate with info-theoretic security (no computational assumptions) using private-key crypto (one-time pad).
Alice
Eve
Bob

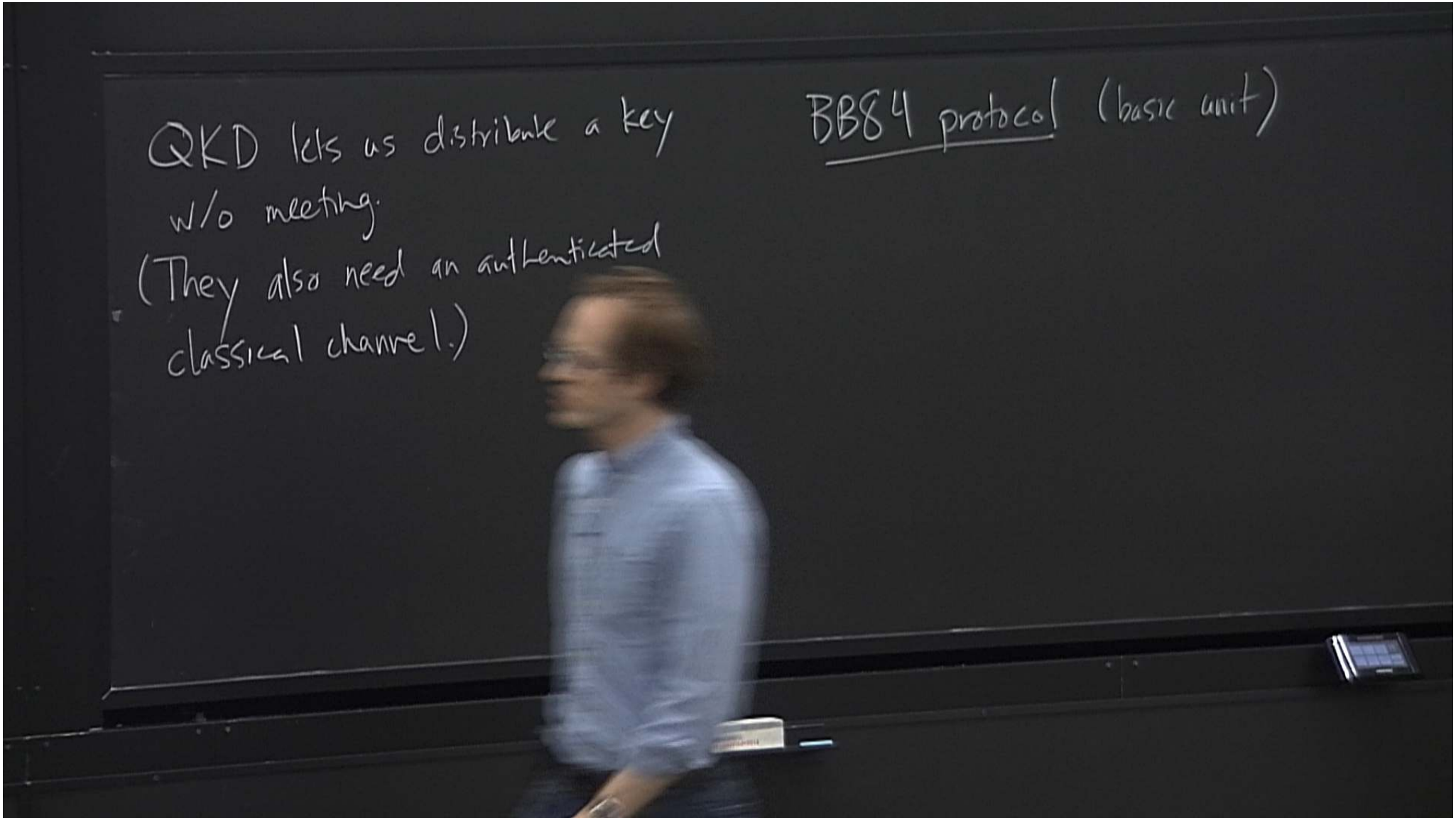

o-theoretic security
ns) using private-key
→ Bob
To send a bit m ∈ {0,1}
A & B share a secret random bit k ∈ {0,1}
A computes m ⊕ k = c
E can get no information about m from c.

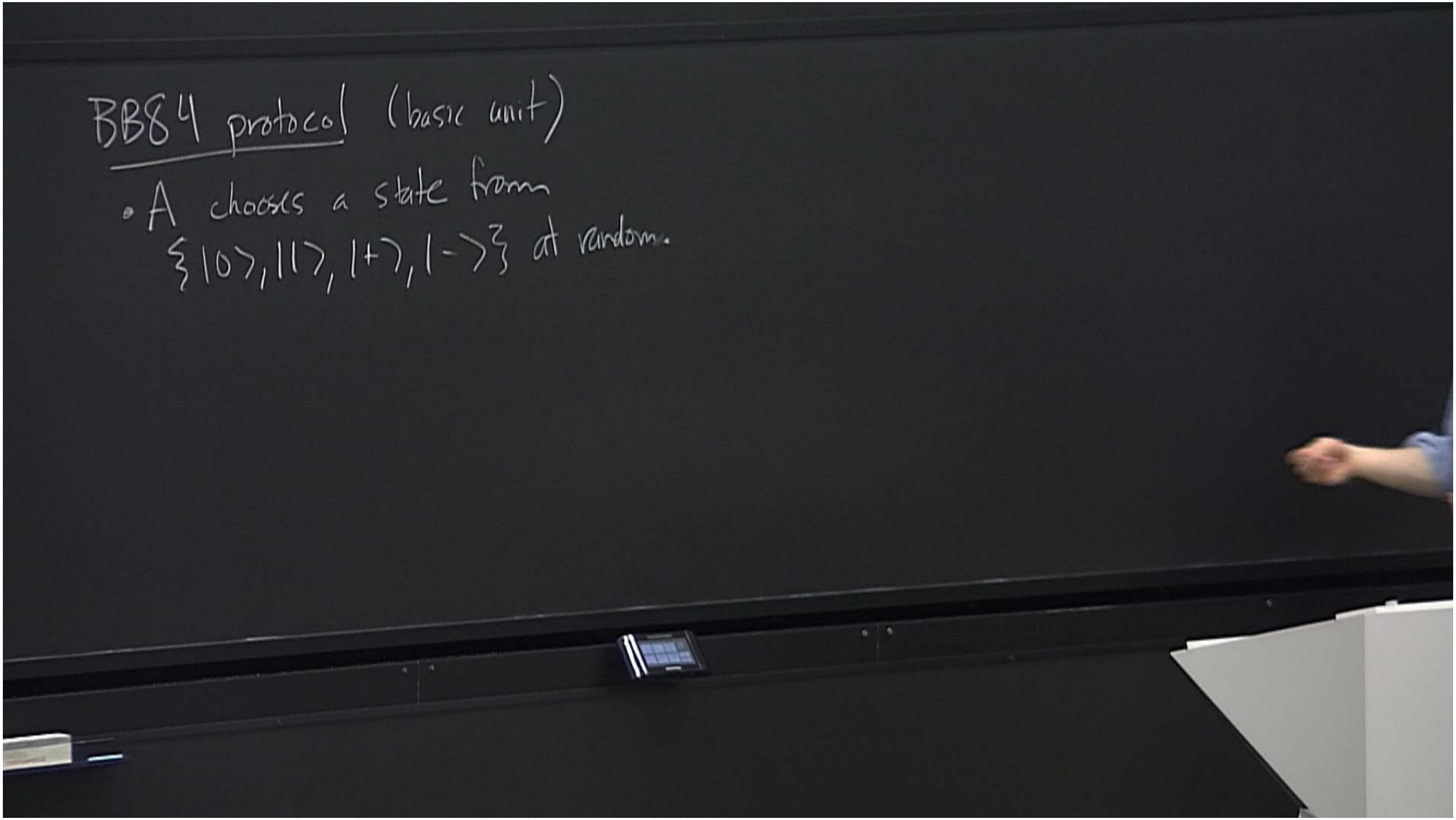

o-theoretic security
ns) using private-key
→ Bob
To send a bit m ∈ {0,1}
A & B share a secret random bit k ∈ {0,1}
A computes m ⊕ k = c
E can get no information about m from c.
B can compute c ⊕ k = m

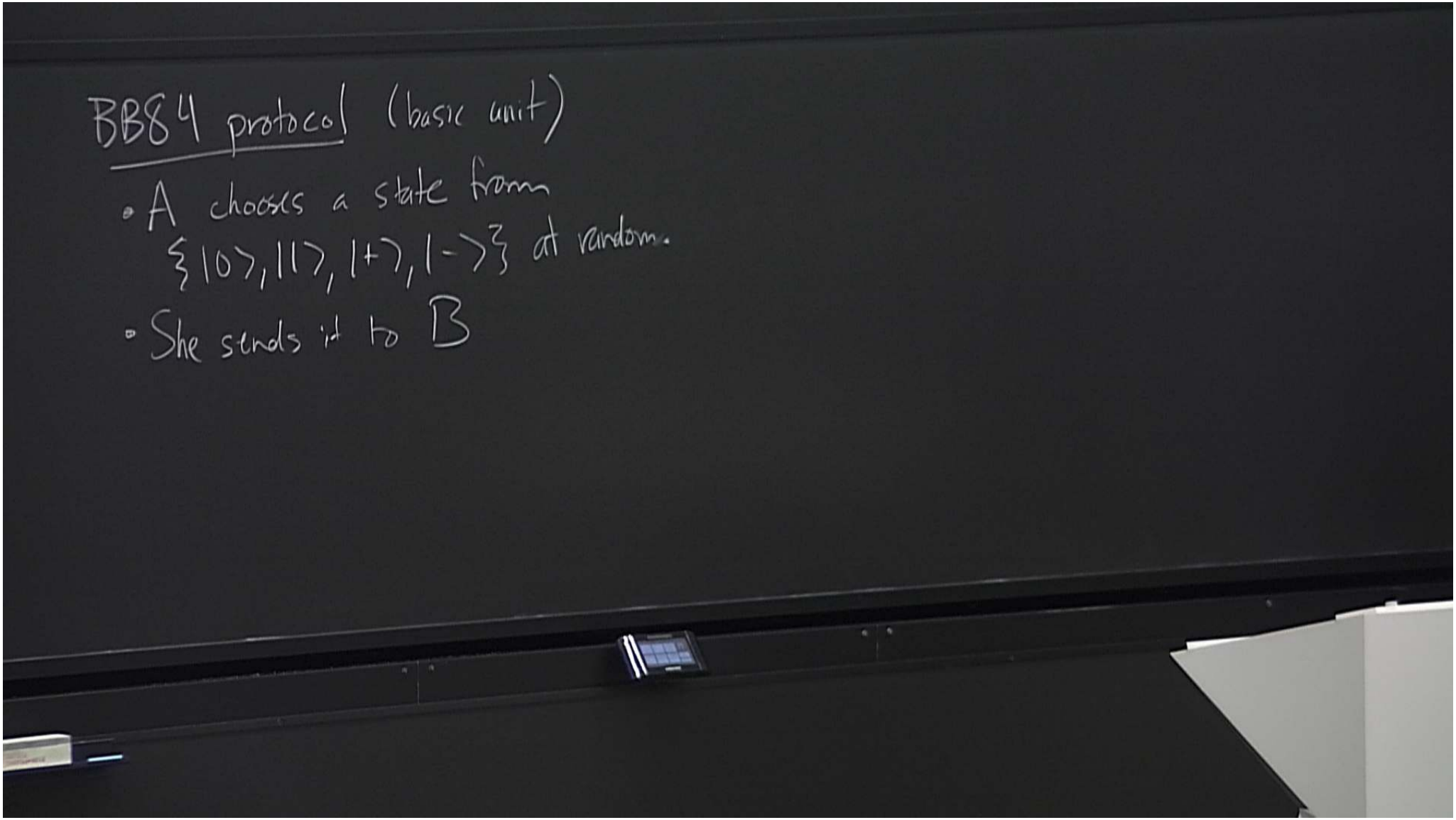

QKD lets us distribute a key
w/o meeting.
(They also need an authenticated
classical channel.)

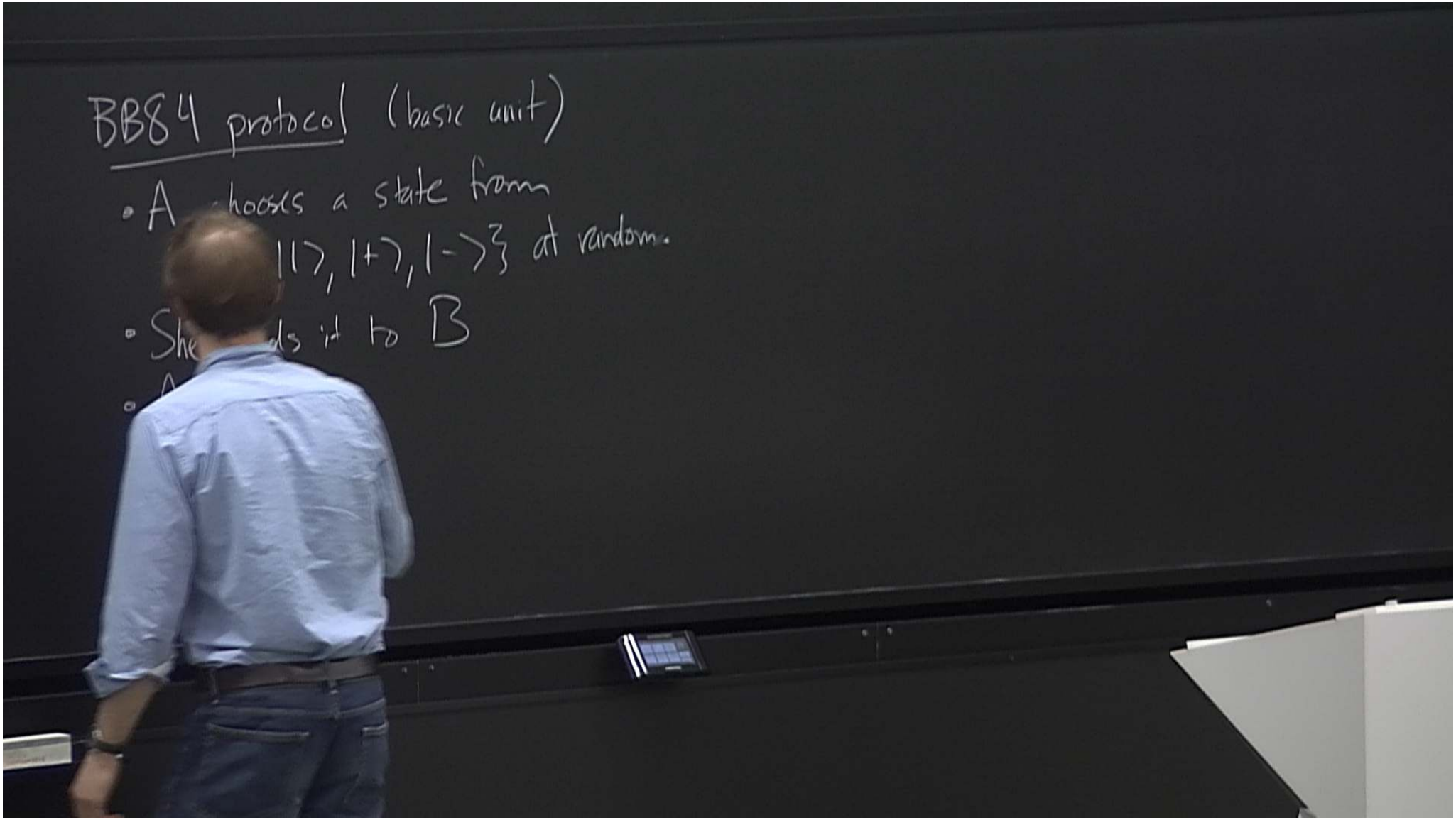

QKD lets us distribute a key
w/o meeting.
(They also need an authenticated
classical channel.)

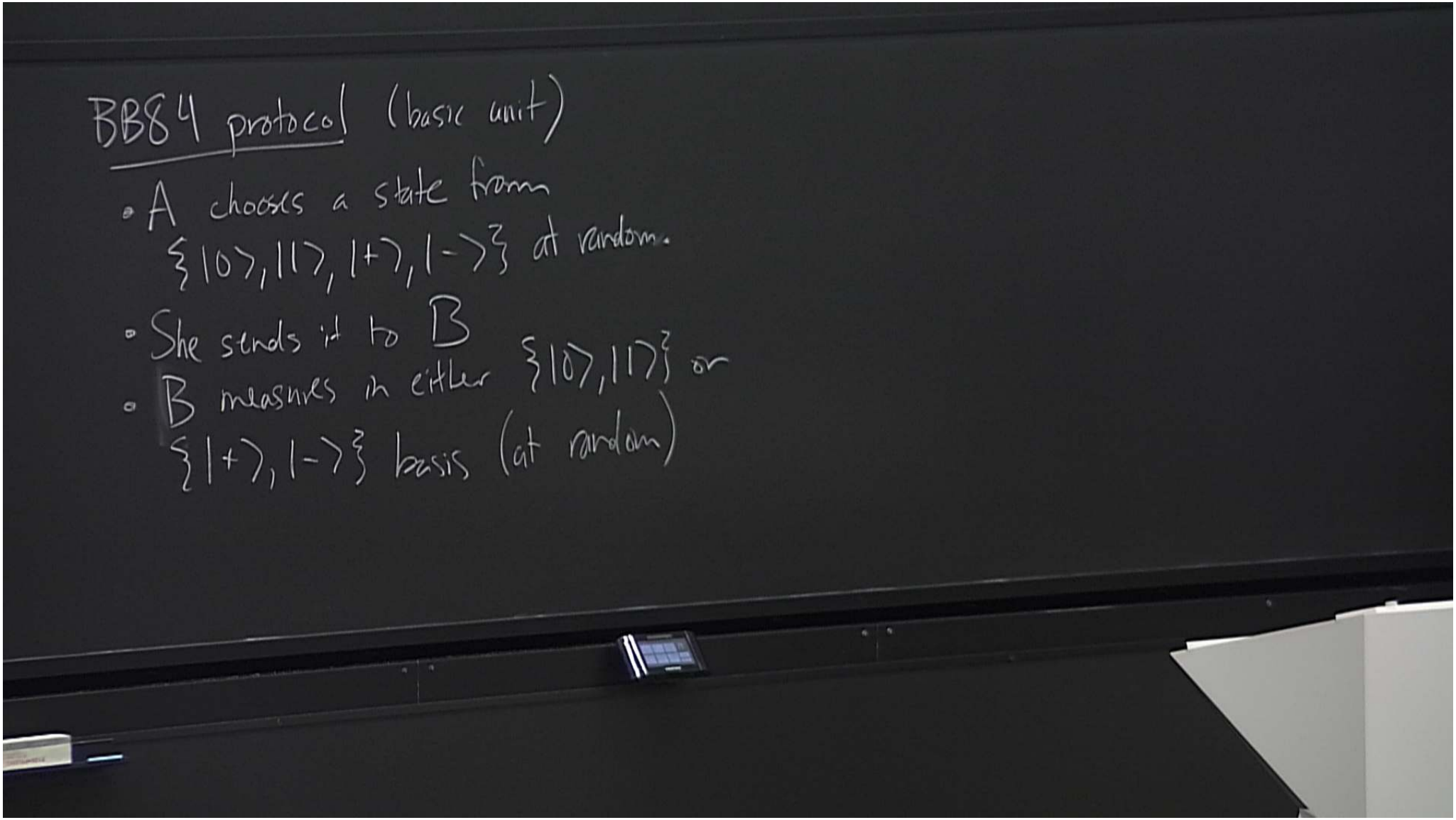

QKD lets us distribute a key
w/o meeting.
(They also need an authenticated
classical channel.)

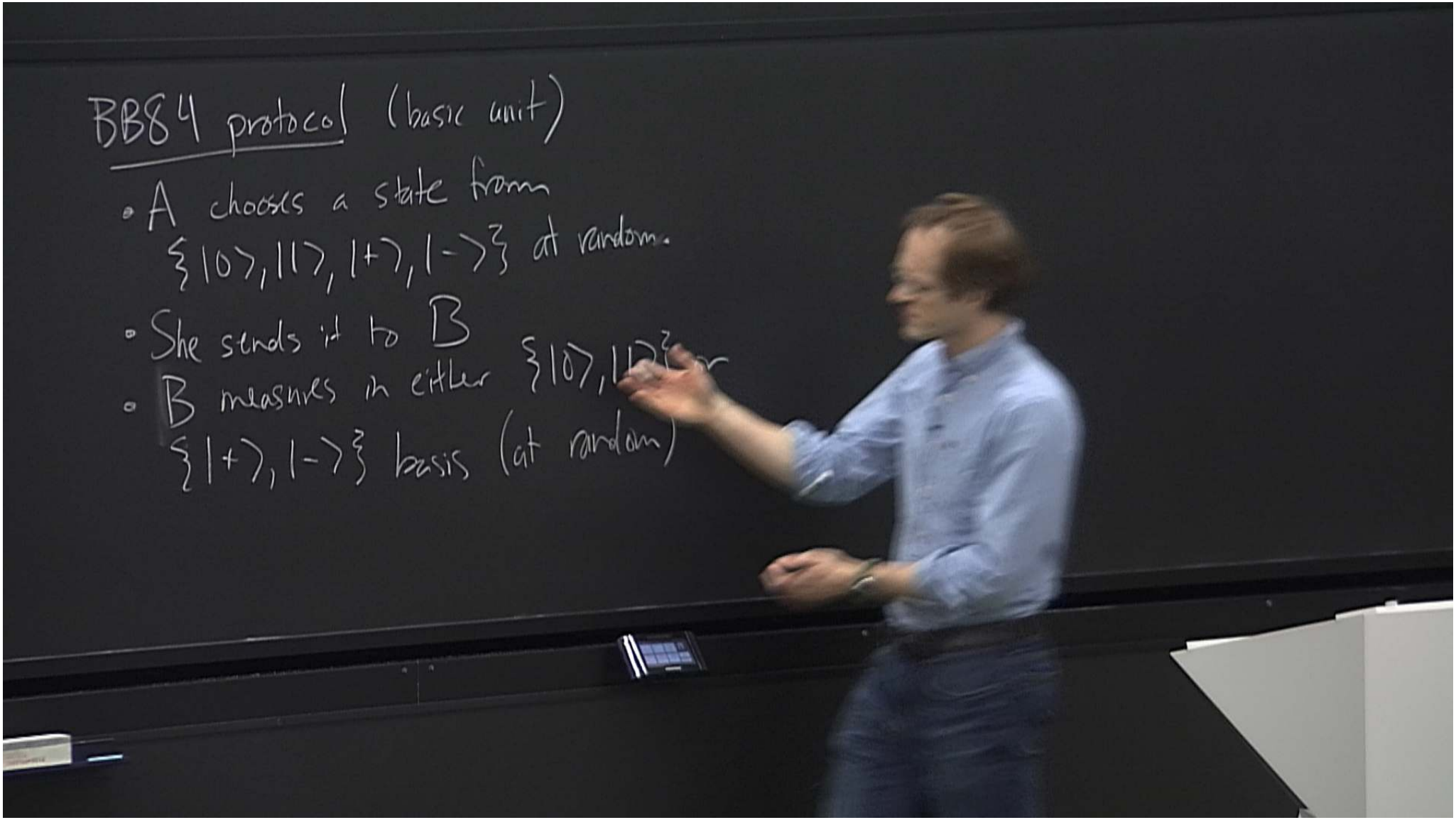

QKD lets us distribute a key
w/o meeting.
(They also need an authenticated
classical channel.)
BB84 protocol (basic unit)

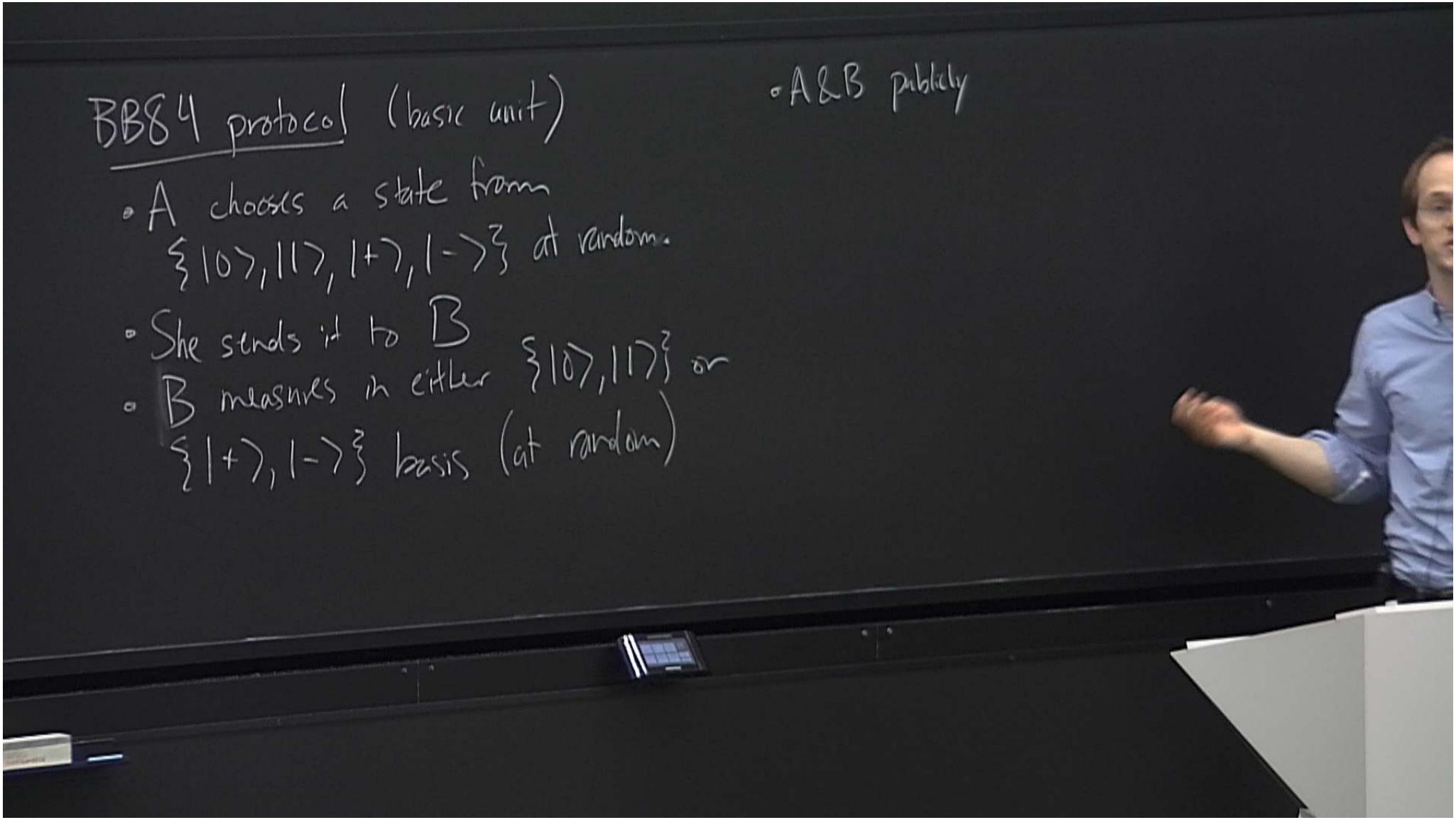

BB84 protocol (basic unit)
• A chooses a state from
{|0⟩, |1⟩, |+⟩, |−⟩} at random.

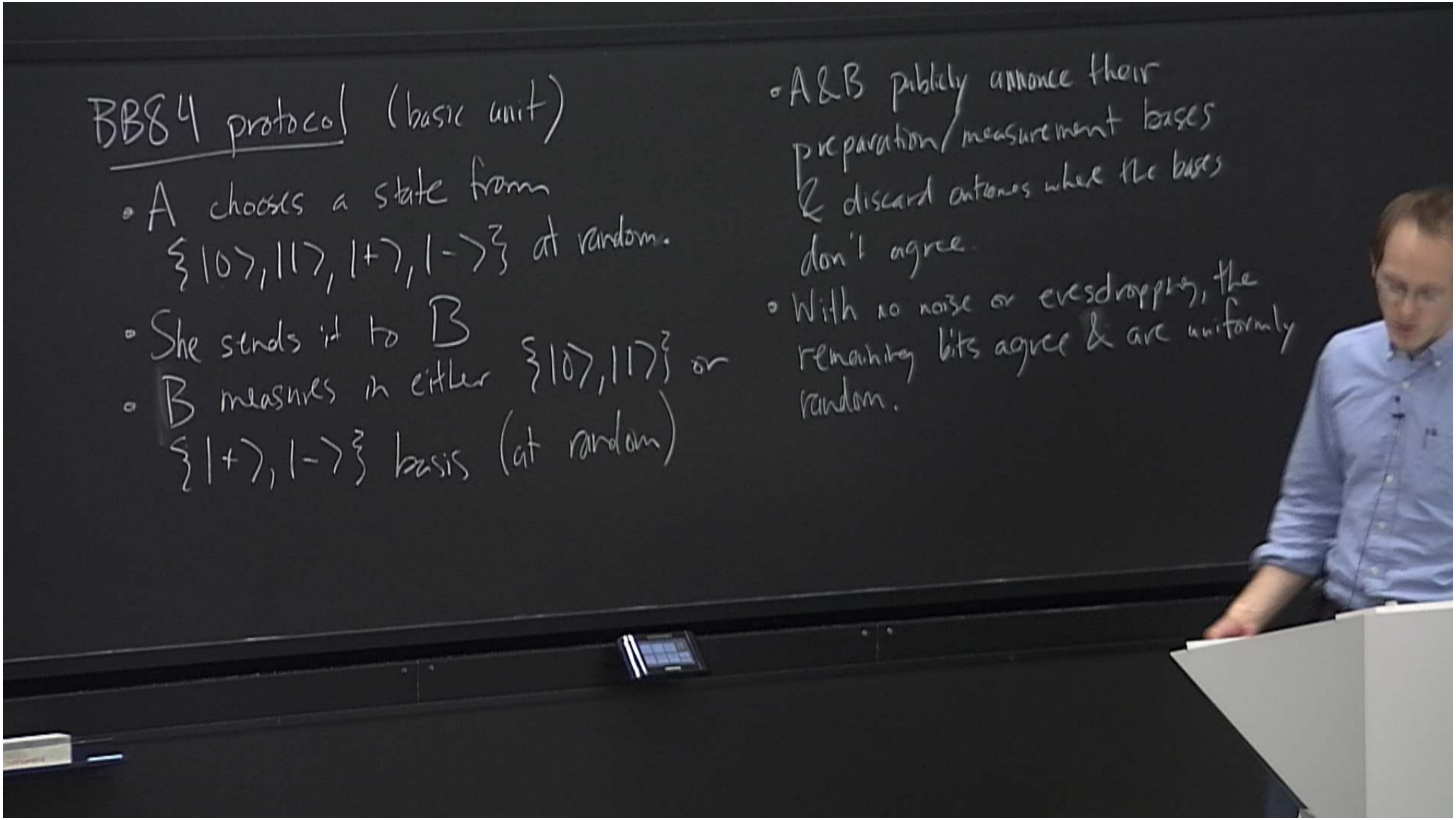

BB84 protocol (basic unit)
• A chooses a state from
{|0⟩, |1⟩, |+⟩, |−⟩} at random.
• She sends it to B

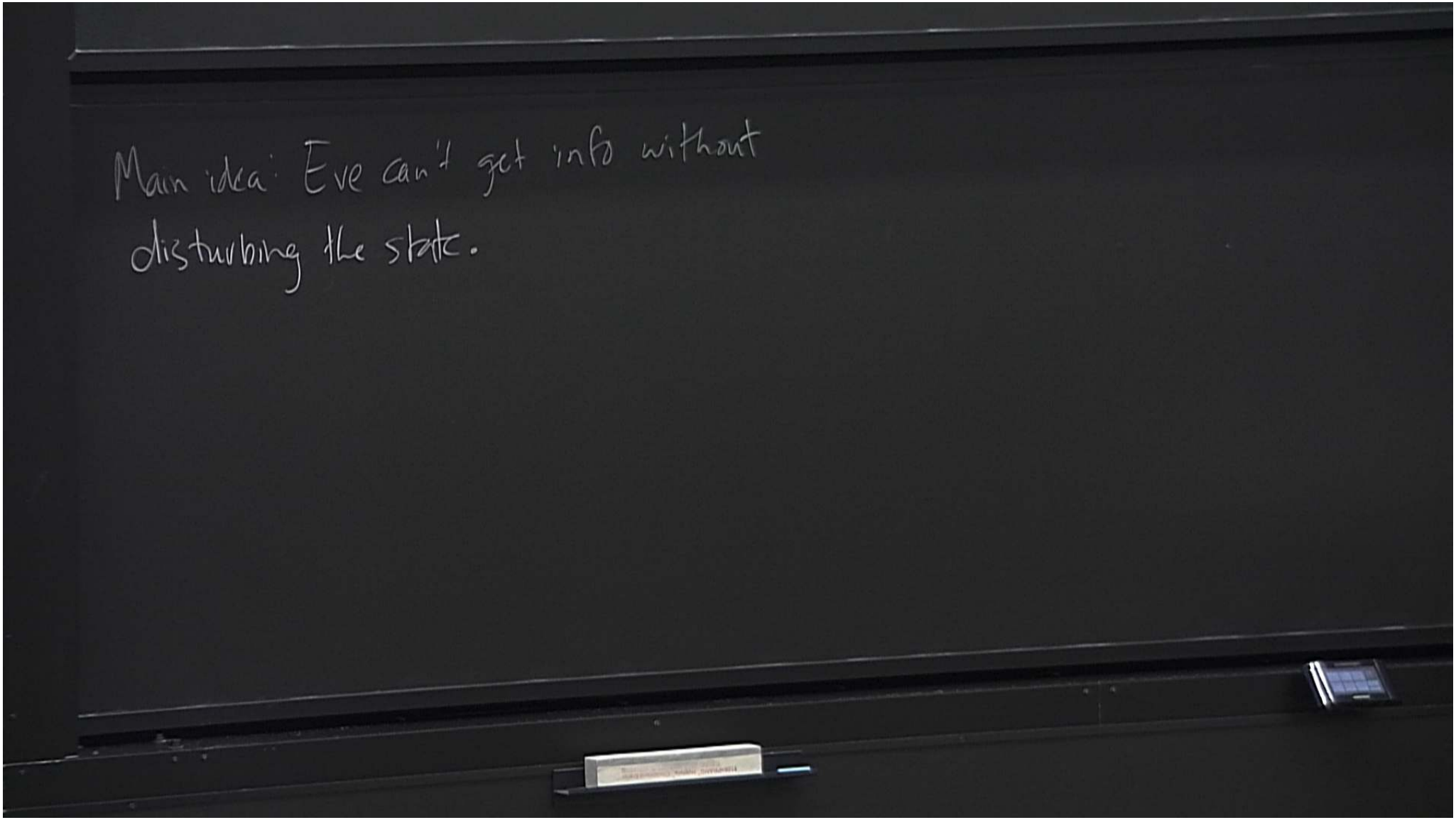

BB84 protocol (basic unit)
• A chooses a state from
|1>, |+>, |->} at random.
• She sends it to B


BB84 protocol (basic unit)
• A chooses a state from
{|0>, |1>, |+>, |->} at random.
• She sends it to B
• B measures in either {|0>, |1>} or
{|+>, |->} basis (at random)


BB84 protocol (basic unit)
• A chooses a state from
{|0>, |1>, |+>, |->} at random.
• She sends it to B
• B measures in either {|0>, |1>}
{|+>, |->} basis (at random)


BB84 protocol (basic unit)
• A chooses a state from
{|0⟩, |1⟩, |+⟩, |−⟩} at random.
• She sends it to B
• B measures in either {|0⟩, |1⟩} or
{|+⟩, |−⟩} basis (at random)
• A & B publicly


BB84 protocol (basic unit)
• A chooses a state from {|0>, |1>, |+>, |->} at random.
• She sends it to B
• B measures in either {|0>, |1>} or {|+>, |->} basis (at random)
• A&B publicly announce their preparation/measurement bases & discard outcomes where the bases don't agree.
• With no noise or eavesdropping, the remaining bits agree & are uniformly random.


Main idea: Eve can't get info without
disturbing the state.

Main idea: Eve can't get info without disturbing the state.

Ex: Intercept-resend attack.

Main idea: Eve can't get info without disturbing the state.

Ex: Intercept-resend attack.

E measures in a random basis

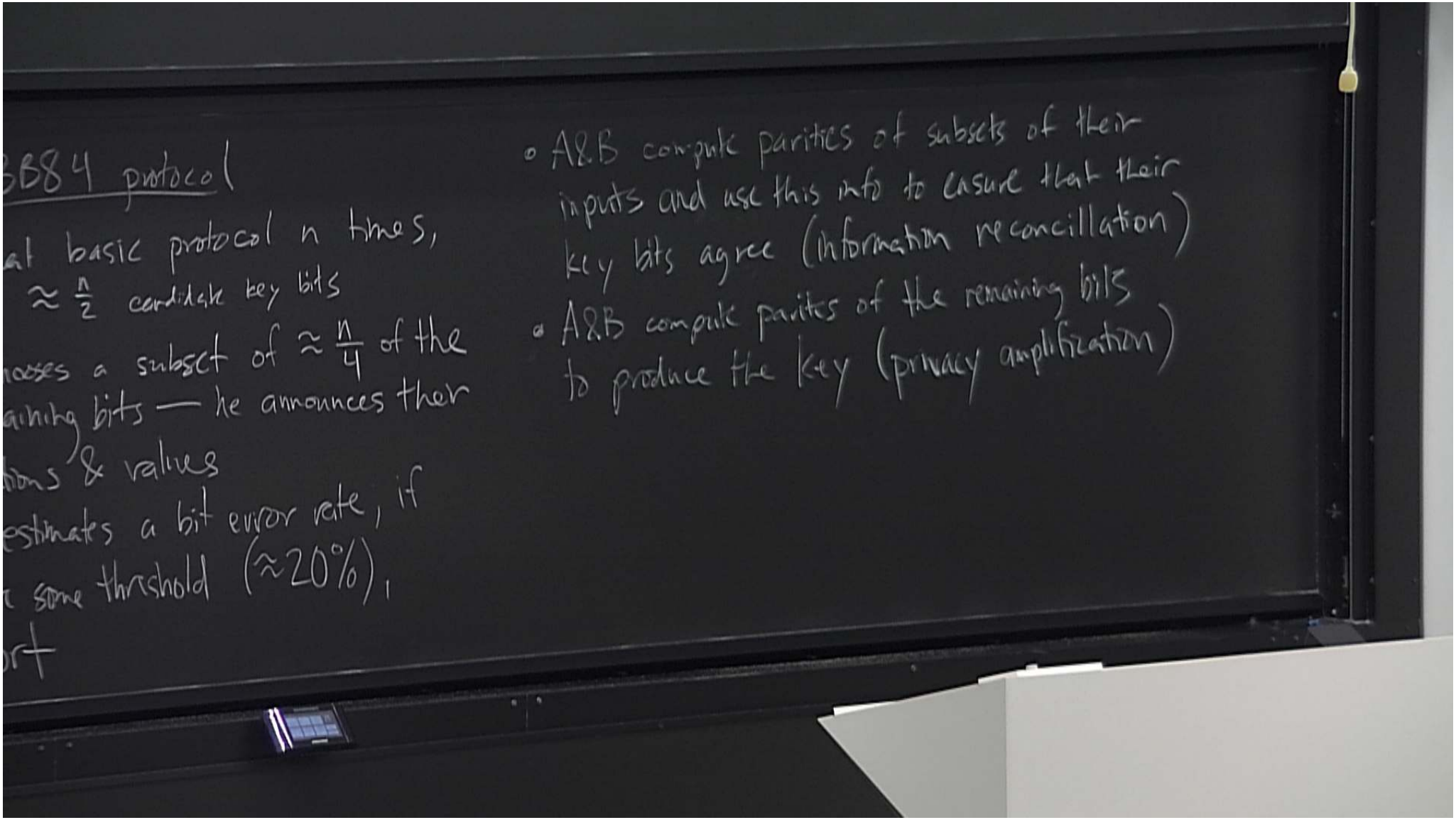
With some probability, she learns a bit of key.

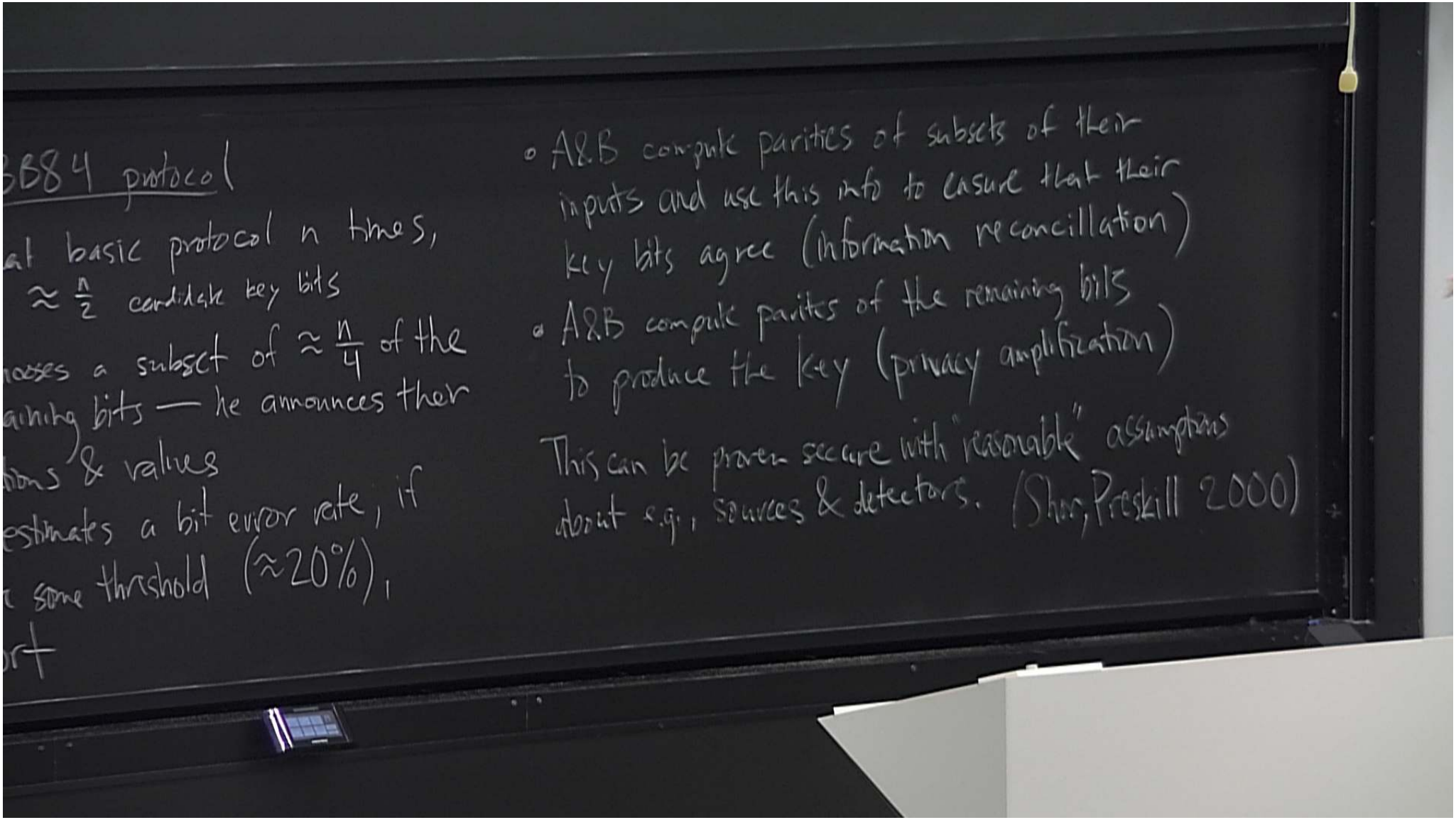
With some (also fairly large) probability, A&B will get bits that should agree but don't.

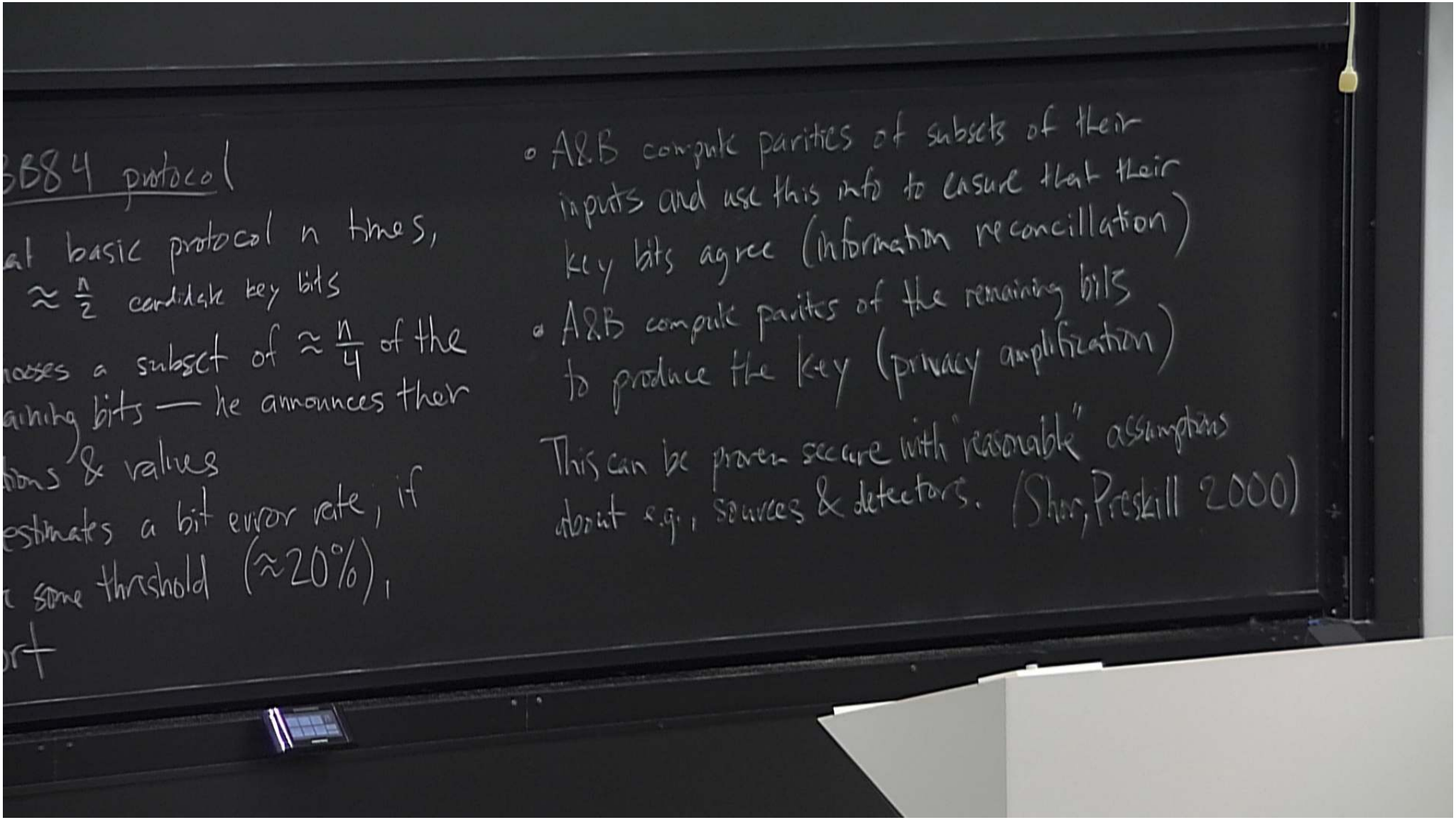
a: Eve can't get info without
oing the state.
Intercept-resend attack.
asures in a random basis
some probability, she learns a
t of key.
some (also fairly large) probability,
B will get bits that should agree but don't.

Full BB84 protocol

- Repeat basic protocol n times, get $\approx \frac{n}{2}$ candidate key bits
- A chooses a subset of $\approx \frac{n}{4}$ of the remaining bits — he announces their locations & values
- B estimates a bit error rate, if above some threshold


BB84 protocol
at basic protocol n times,
≈ n/2 candidate key bits
noses a subset of ≈ n/4 of the
aining bits — he announces ther
ions & values
estimates a bit error rate, if
some threshold (≈20%),
ort
• A&B compute parities of subsets of their inputs and use this info to ensure that their key bits agree (information reconcillation)
• A&B compute parites of the remaining bits to produce the key (privacy amplification)


BB84 protocol
at basic protocol n times,
≈ n/2 candidate key bits
asses a subset of ≈ n/4 of the
aining bits — he announces ther
ions & values
estimates a bit error rate, if
some threshold (≈20%),
ort
• A&B compute parities of subsets of their inputs and use this info to ensure that their key bits agree (information reconcillation)
• A&B compute parites of the remaining bits to produce the key (privacy amplification)
This can be proven secure with "reasonable" assumptions about e.g., sources & detectors. (Shor, Preskill 2000)


BB84 protocol
at basic protocol n times,
≈ n/2 candidate key bits
passes a subset of ≈ n/4 of the
aining bits — he announces their
ions & values
estimates a bit error rate, if
some threshold (≈20%),
ort
• A&B compute parities of subsets of their inputs and use this info to ensure that their key bits agree (information reconcillation)
• A&B compute parites of the remaining bits to produce the key (privacy amplification)
This can be proven secure with "reasonable" assumptions about e.g., sources & detectors. (Shor, Preskill 2000)