A Novel Approach to Dual Execution for YGC With Applications to Fuzzy PAKE

Sophia Yakoubov

joint work with

Pierre-Alain Dupont, Julia Hesse, David Pointcheval, Leonid Reyzin

Published at Eurocrypt 2018

A Novel Approach

- Our contribution: eliminating this leakage…
  - in a limited, interesting setting

To Dual Execution

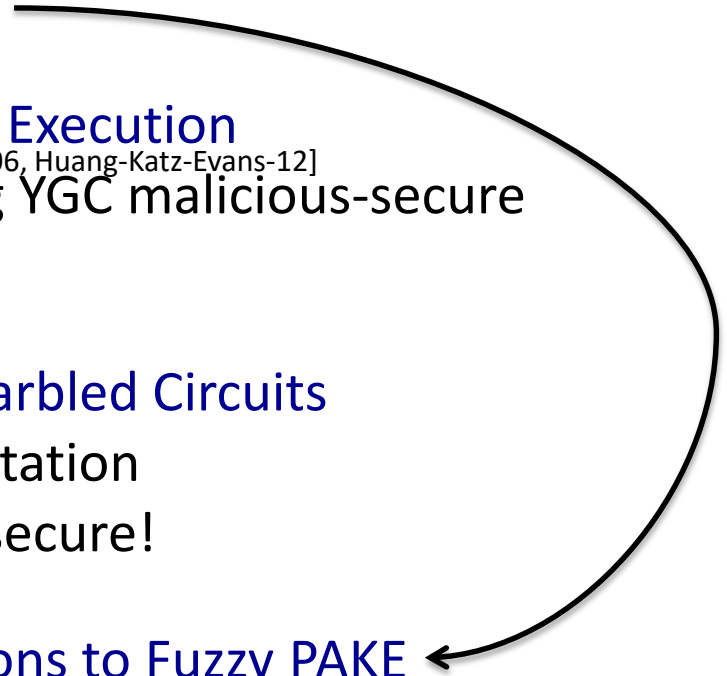
[Mohassel-Franklin-06, Huang-Katz-Evans-12]

- Efficient transformation making YGC malicious-secure
- Downside: it leaks a bit!

For Yao's Garbled Circuits

- YGC: efficient two-party computation
- Problem: YGC is not malicious-secure!

With Applications to Fuzzy PAKE



A Novel Approach

- Our contribution: eliminating this leakage…
  - in a limited, interesting setting

To Dual Execution

[Mohassel-Franklin-06, Huang-Katz-Evans-12]

- Efficient transformation making YGC malicious-secure
- Downside: it leaks a bit!

For Yao's Garbled Circuits

- YGC: efficient two-party computation
- Problem: YGC is not malicious-secure!

With Applications to **Fuzzy PAKE**

Motivation

p@$$w0rd12

- Want: secure communication
- Over insecure, unauthenticated channel
- Shared secret: password
- The password is…
  - Low-entropy

p@$$w0rd12

Motivation

p@$$w0rd12

- Want: secure communication
- Over insecure, unauthenticated channel
- Shared secret: password
- The password is…
  - Low-entropy
  - Possibly noisy

p@$$w@rd12

Motivation

p@$$w0rd12

- Goal: Agree on high-entropy cryptographic key
- Man-in-the-middle security: Nothing leaks about…
  - Password
  - Key

p@$$w@rd12

Applications

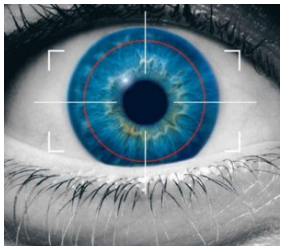
p@$$w0rd12

- Mistyped passwords

  e.g. [Chatterjee-Athalye-Akhawe-Juels-Ristenpart-16]

p@$$w@rd12

Applications: Not Just Passwords!



- Mistyped passwords
- Biometric authentication

Bob has a resource Alice is trying to access

Applications: Not Just Passwords!

- Mistyped passwords
- Biometric authentication
- Location-based authentication
  e.g. [Han-Harishankar-Wang-Chung-Tague-17]

"radiation stinks has 3 potholes"

"radiator springs has 4 potholes"

Related Work

are the passwords **low-entropy**?
low-entropy: can hit correct password by brute-force enumeration

do the passwords have **noise**?

Related Work

are the passwords **low-entropy**?

do the passwords have **noise**?

	Low-entropy password	High-entropy password
Exact match		
Fuzzy match		

Related Work

	Low-entropy password	High-entropy password
Exact match		privacy amplification [Maurer-97, …]
Fuzzy match		

Related Work

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match		privacy amplification [Maurer-97, …]
Fuzzy match		

Related Work

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match	PAKE [Bellare-Pointcheval-Rogaway-00, Boyko-MacKenzie-Patel-00, …]	privacy amplification [Maurer-97, …]
Fuzzy match		

Secure against off-line dictionary attacks against the password

Related Work

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match	PAKE [Bellare-Pointcheval-Rogaway-00, Boyko-MacKenzie-Patel-00, …]	privacy amplification [Maurer-97, …]
Fuzzy match		information reconciliation [Renner-Wolf-04, …] robust fuzzy extractors [Boyen-Dodis-Katz-Ostrovsky-Smith-05, …]

Related Work

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match	PAKE [Bellare-Pointcheval-Rogaway-00, Boyko-MacKenzie-Patel-00, …]	privacy amplification [Maurer-97, …]
Fuzzy match	?	information reconciliation [Renner-Wolf-04, …] robust fuzzy extractors [Boyen-Dodis-Katz-Ostrovsky-Smith-05, …]

Fuzzy PAKE

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match	PAKE [Bellare-Pointcheval-Rogaway-00, Boyko-MacKenzie-Patel-00, …]	privacy amplification [Maurer-97, …]
Fuzzy match	New Primitive - Fuzzy PAKE	information reconciliation [Renner-Wolf-04, …] robust fuzzy extractors [Boyen-Dodis-Katz-Ostrovsky-Smith-05, …]

Fuzzy PAKE

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match	PAKE [Bellare-Pointcheval-Rogaway-00, Boyko-MacKenzie-Patel-00, …]	privacy amplification [Maurer-97, …]
Fuzzy match	New Primitive - Fuzzy PAKE	information reconciliation [Renner-Wolf-04, …] robust fuzzy extractors [Boyen-Dodis-Katz-Ostrovsky-Smith-05, …]

Our Contributions

- Security definition
- Efficient constructions

of Fuzzy Password Authenticated Key Exchange

Fuzzy PAKE

	Low-entropy password (no leakage allowed)	High-entropy password (some leakage ok)
Exact match	PAKE [Bellare-Pointcheval-Rogaway-00, Boyko-MacKenzie-Patel-00, …]	privacy amplification [Maurer-97, …]
Fuzzy match	New Primitive - Fuzzy PAKE	information reconciliation [Renner-Wolf-04, …] robust fuzzy extractors [Boyen-Dodis-Katz-Ostrovsky-Smith-05, …]

Our Contributions

- Security definition
- Efficient constructions

of Fuzzy Password Authenticated Key Exchange

What to do?

p@$$w0rd12

p@$$w@rd12

Use MPC!

p@$$w0rd12

p@$$w@rd12

- Problem: unauthenticated channels!

Use MPC!

p@$$w0rd12

p@$$w@rd12

- Problem: unauthenticated channels!
- Solution: secure computation without authentication [Barak-Canetti-Lindell-Pass-Rabin-05]
  - Generic transformation for MPC*
  - Cheap: just add digital signatures (without PKI)!

*I'm skipping a bunch of details

Use MPC!

p@$$w0rd12

p@$$w@rd12

- Problem: unauthenticated channels!
- Solution: secure computation without authentication [Barak-Canetti-Lindell-Pass-Rabin-05]
- Q: Which MPC?
- A: Yao's Garbled Circuits!

A Novel Approach

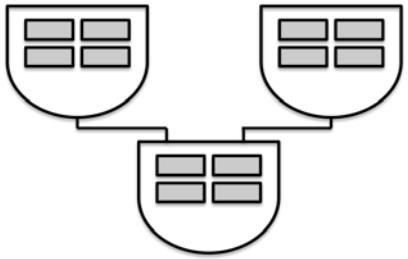
To Dual Execution

For **Yao's Garbled Circuits**

With Applications to Fuzzy PAKE

Yao's Garbled Circuits for FPAKE

garbler



evaluator

output

Yao's Garbled Circuits for FPAKE

pw_G = p@$$w0rd12

pw_E = p@$$w@rd12

Circuit determining whether pw_G and pw_E are "close enough"; outputs session key if yes

garbler

evaluator

output

Yao's Garbled Circuits for FPAKE

pw_G = p@$$w0rd12

pw_E = p@$$w@rd12

Circuit determining whether pw_G and pw_E are "close enough"; outputs session key if yes

output

semi-honest garbler

malicious evaluator

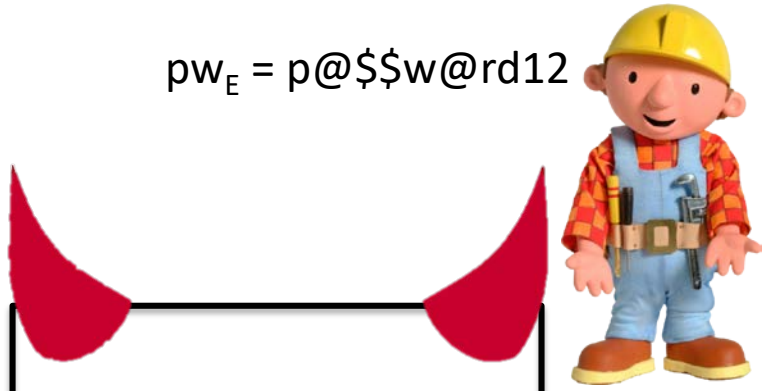
Yao's Garbled Circuits are an asymmetric 2PC protocol: they are secure against a malicious evaluator, but only against a semi-honest garbler

Yao's Garbled Circuits for FPAKE

pw_G = p@$$w0rd12

~~semi-honest~~
garbler

pw_E = p@$$w@rd12



malicious
evaluator

Yao's Garbled Circuits are an asymmetric 2PC protocol: they are secure against a malicious evaluator, but only against a semi-honest garbler

Yao's Garbled Circuits for FPAKE

pw_G = p@$$w0rd12

~~semi-honest~~ garbler

pw_E = p@$$w@rd12

Circuit that returns the key if pw_E starts with 1!

malicious evaluator

output

Yao's Garbled Circuits are an asymmetric 2PC protocol: they are secure against a malicious evaluator, but only against a semi-honest garbler

From Semi-Honest to Malicious

Correctness	Privacy	Computation Overhead

From Semi-Honest to Malicious

Transformation	Correctness	Privacy	Computation Overhead
None			
Commit-and-Prove			
Cut-and-Choose			
LEGO			(including pre-processing)
…			

- Transformations gain efficiency using…
  - Amortization
  - Pre-processing
- We can't afford either!

From Semi-Honest to Malicious

Transformation	Correctness	Privacy	Computation Overhead
None			
Commit-and-Prove			
Cut-and-Choose			
LEGO			(including pre-processing)
…			
Dual Execution [Mohassel-Franklin-06, Huang-Katz-Evans-12]		1 bit leakage	Only 2x! (+ constant)

1 bit of leakage about a low-entropy password can be crucial!

A Novel Approach

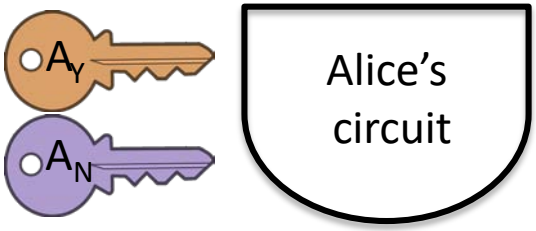
To **Dual Execution**
(for yes-no circuits)

For Yao's Garbled Circuits

With Applications to Fuzzy PAKE

Dual Execution

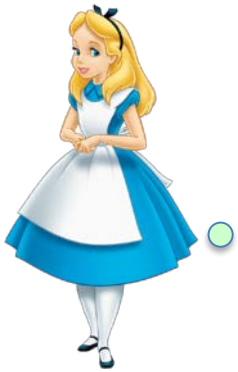
[Mohassel-Franklin-06, Huang-Katz-Evans-12]

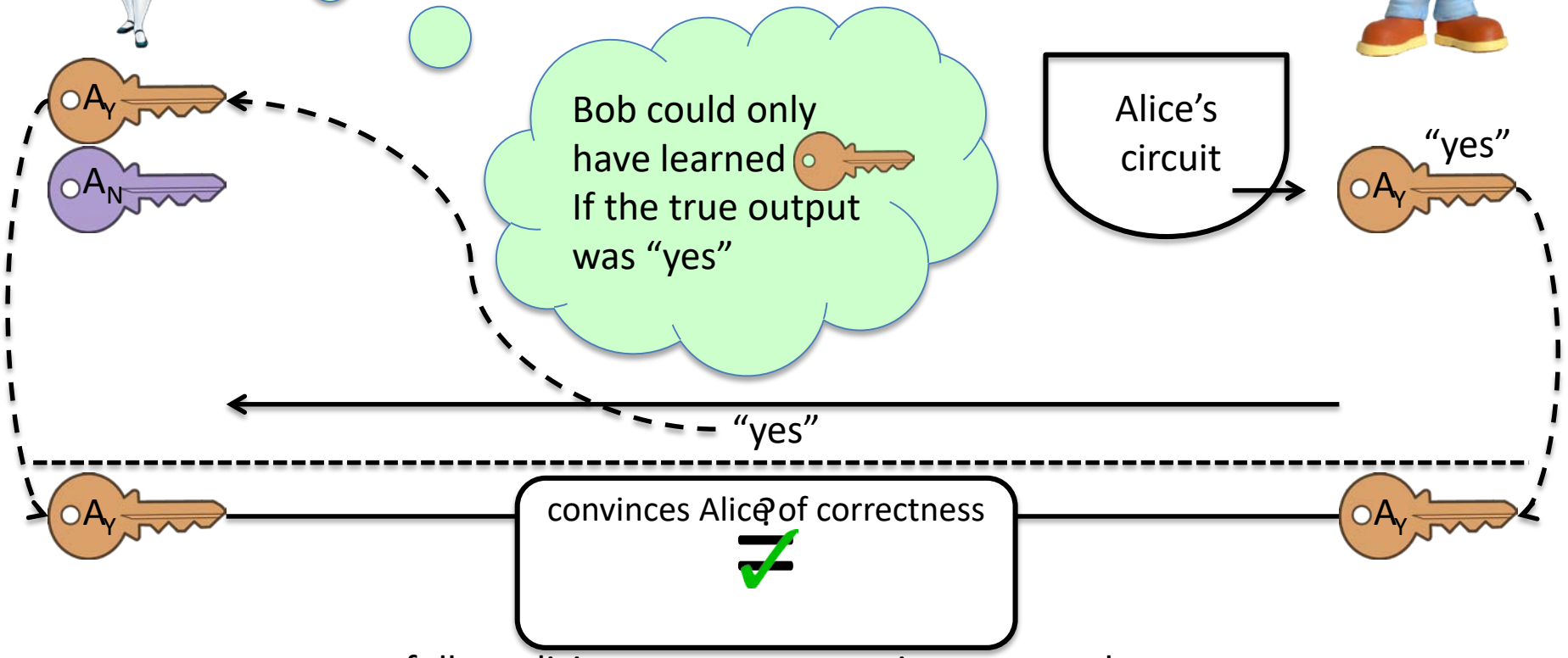


circuit that outputs yes/no and a label

Dual Execution

[Mohassel-Franklin-06, Huang-Katz-Evans-12]

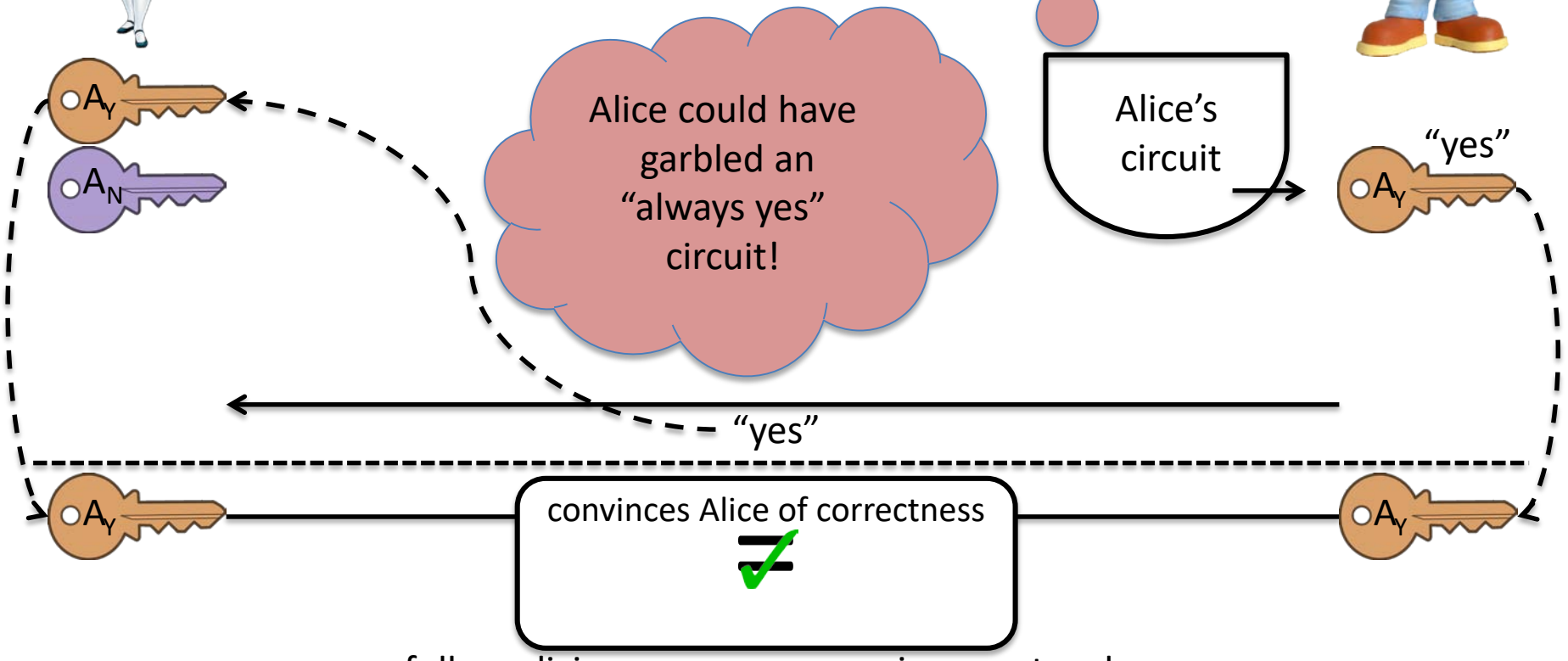




fully malicious-secure comparison protocol

Dual Execution

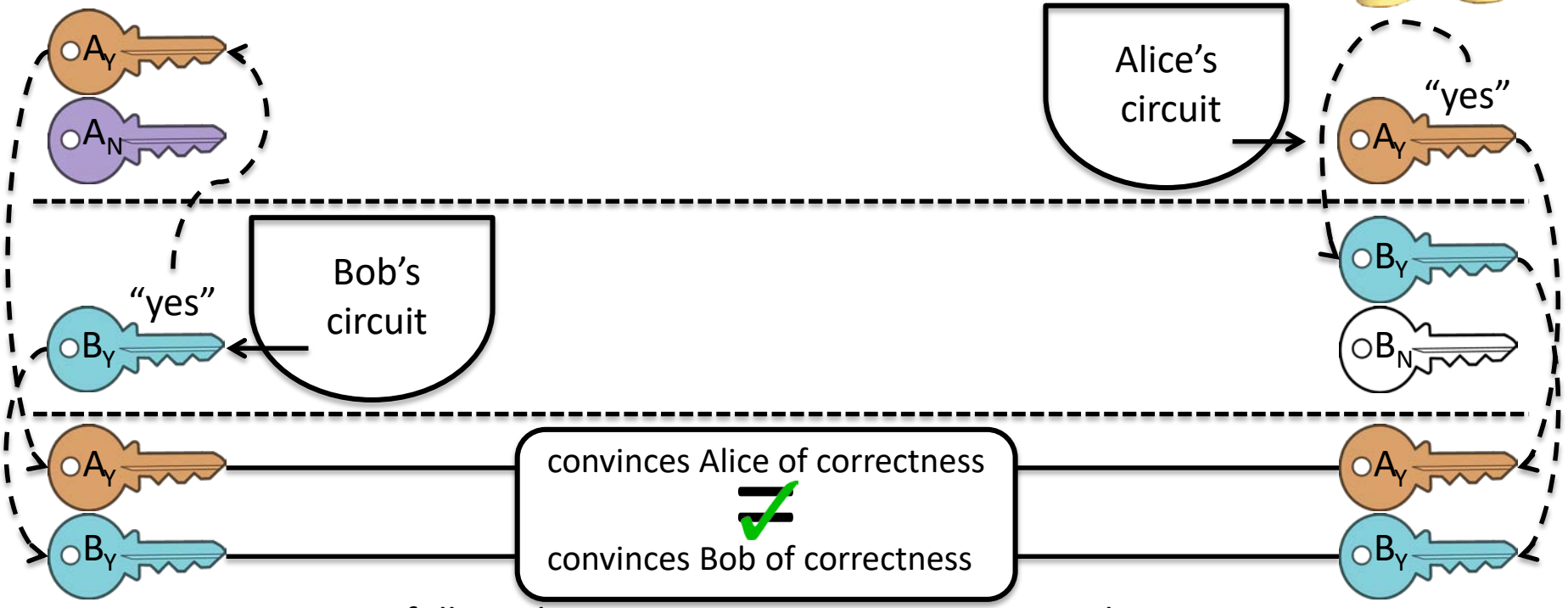
[Mohassel-Franklin-06, Huang-Katz-Evans-12]



fully malicious-secure comparison protocol

Dual Execution

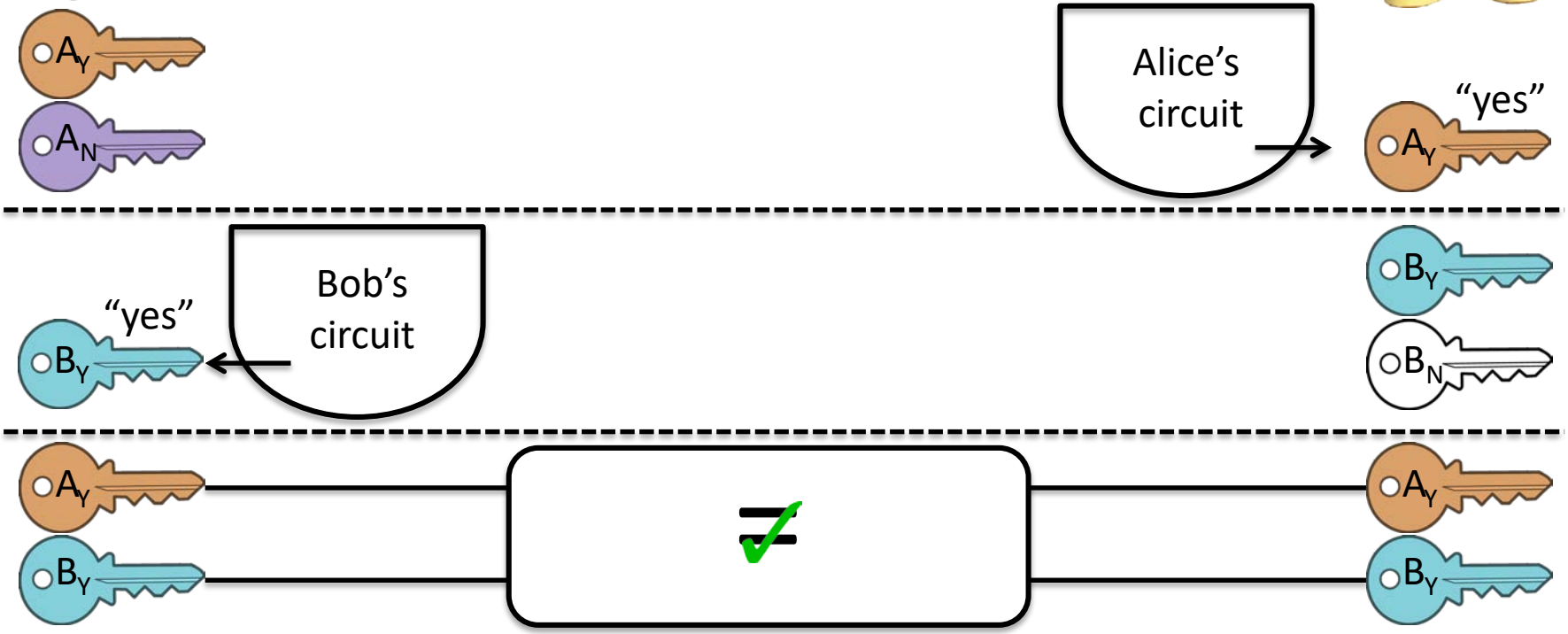
[Mohassel-Franklin-06, Huang-Katz-Evans-12]



fully malicious-secure comparison protocol

Dual Execution

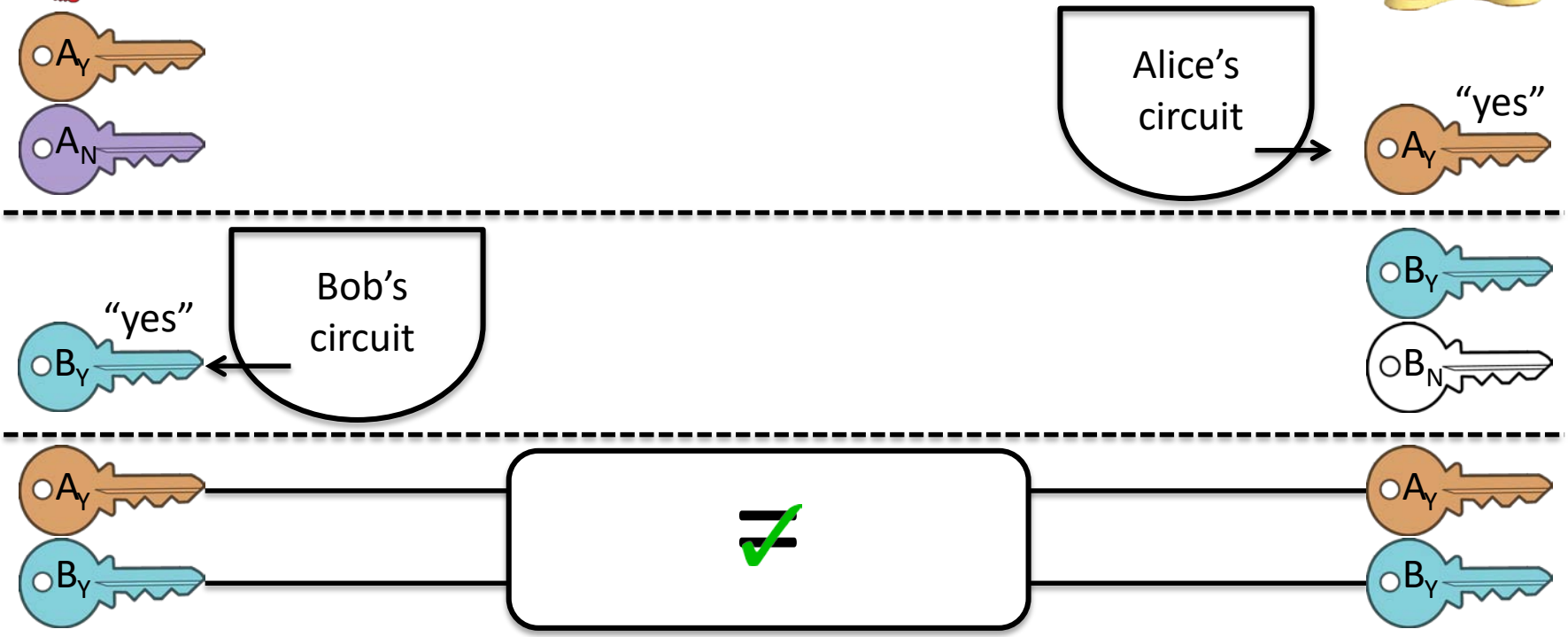
[Mohassel-Franklin-06, Huang-Katz-Evans-12]



[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"		
	"no"		

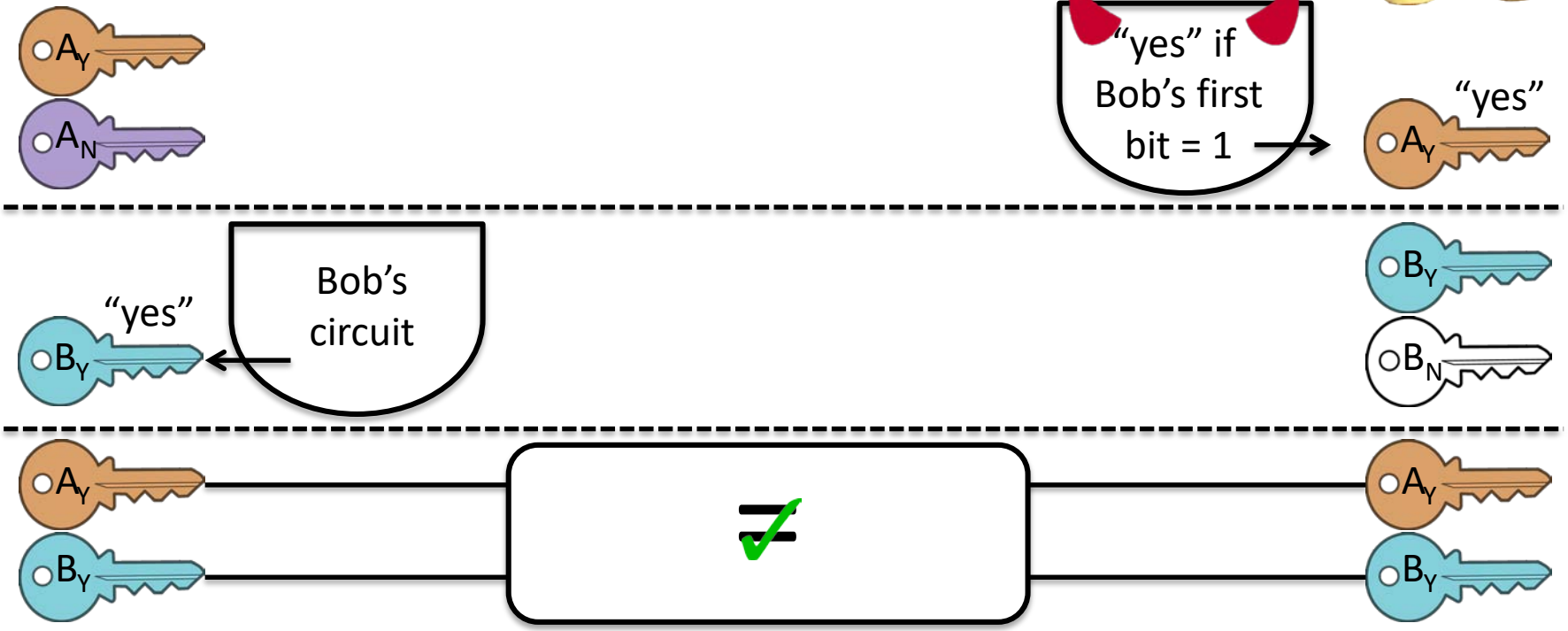
Dual Execution

[Mohassel-Franklin-06, Huang-Katz-Evans-12]



[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "cheating"	
	"no"	"no" or "cheating"	

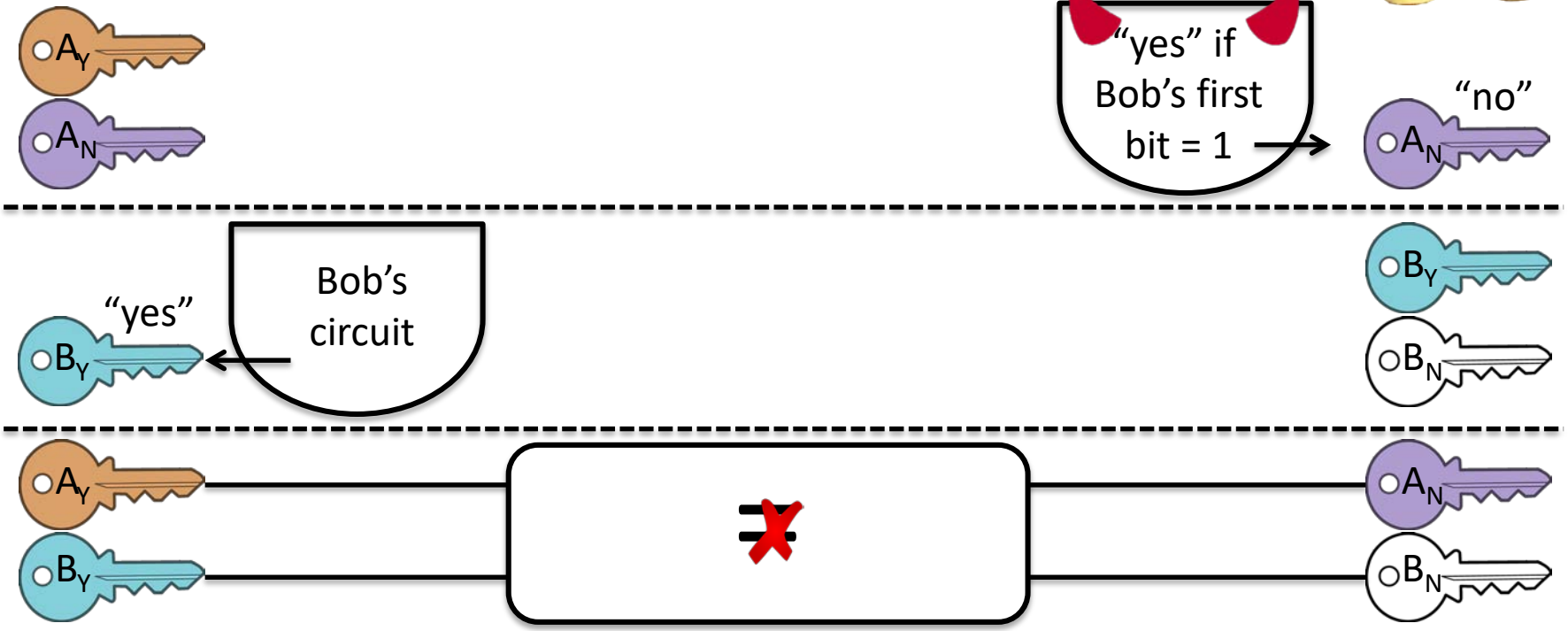
Dual Execution: 1-Bit Leakage Attack



[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "cheating"	1-bit leakage
	"no"	"no" or "cheating"	

If Bob's first bit is 1, equality will hold

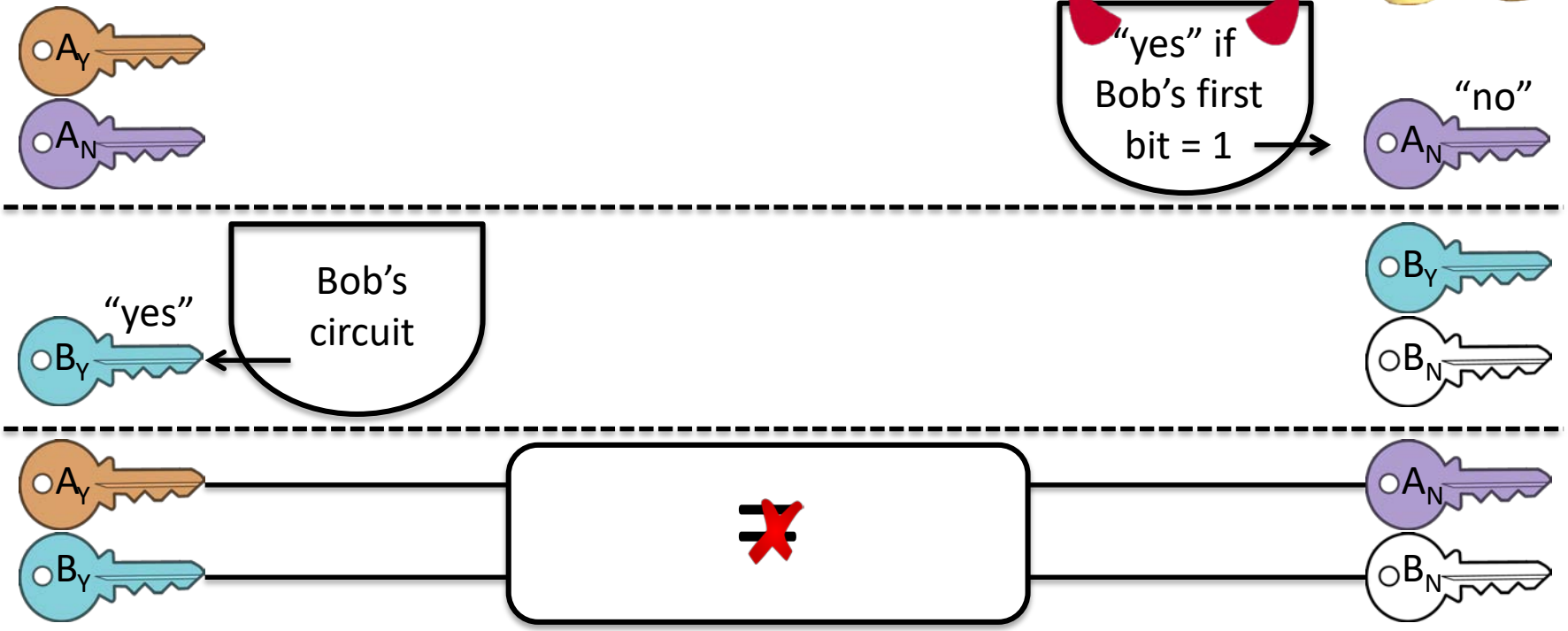
Dual Execution: 1-Bit Leakage Attack



[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "cheating"	1-bit leakage
	"no"	"no" or "cheating"	

If Bob's first bit is 0, equality will fail

Dual Execution: 1-Bit Leakage Attack



[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "cheating"	1-bit leakage
	"no"	"no" or "cheating"	1-bit leakage

If Bob's first bit is 0, equality will fail

A Novel Approach

To Dual Execution

For Yao's Garbled Circuits

With Applications to Fuzzy PAKE

Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions

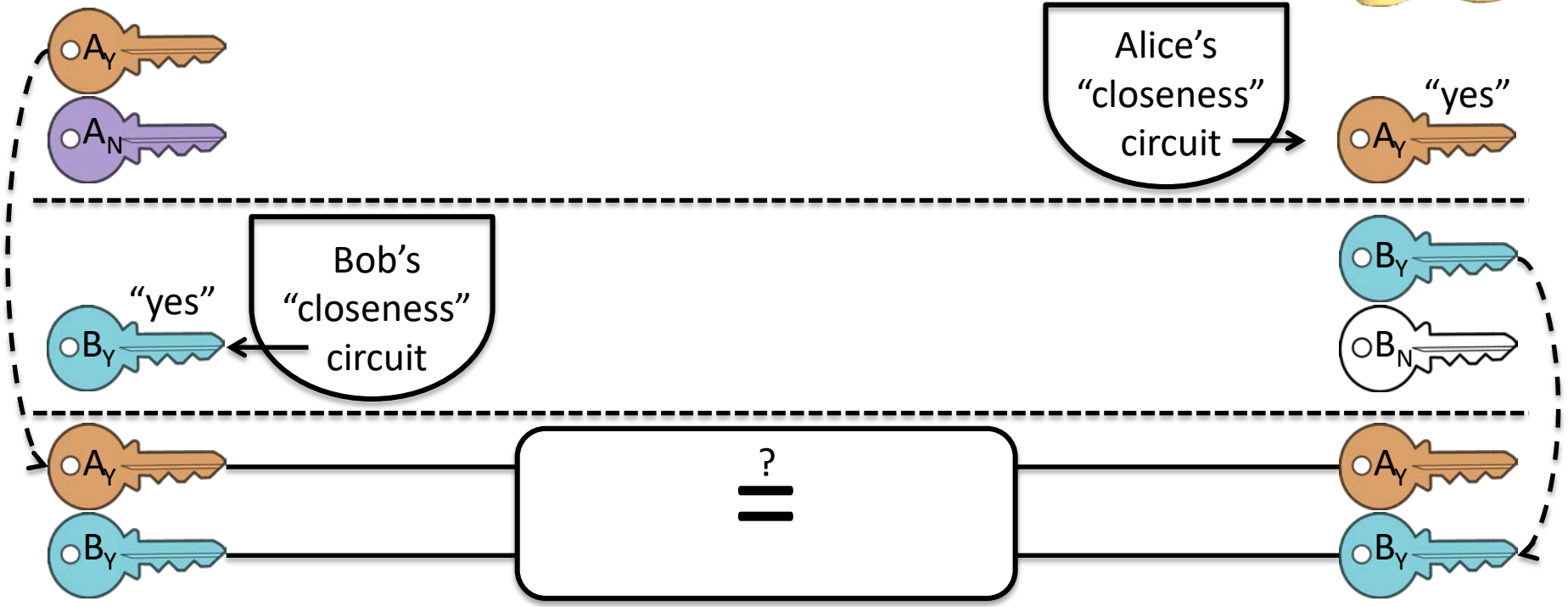
[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "cheating"	1-bit leakage
	"no"	"no" or "cheating"	1-bit leakage

Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "no"	1-bit leakage
	"no"	"no"	yes

This is the perfect tradeoff for fuzzy PAKE!

- Only care about security against adversary who doesn't know a close-enough password – the "no" case

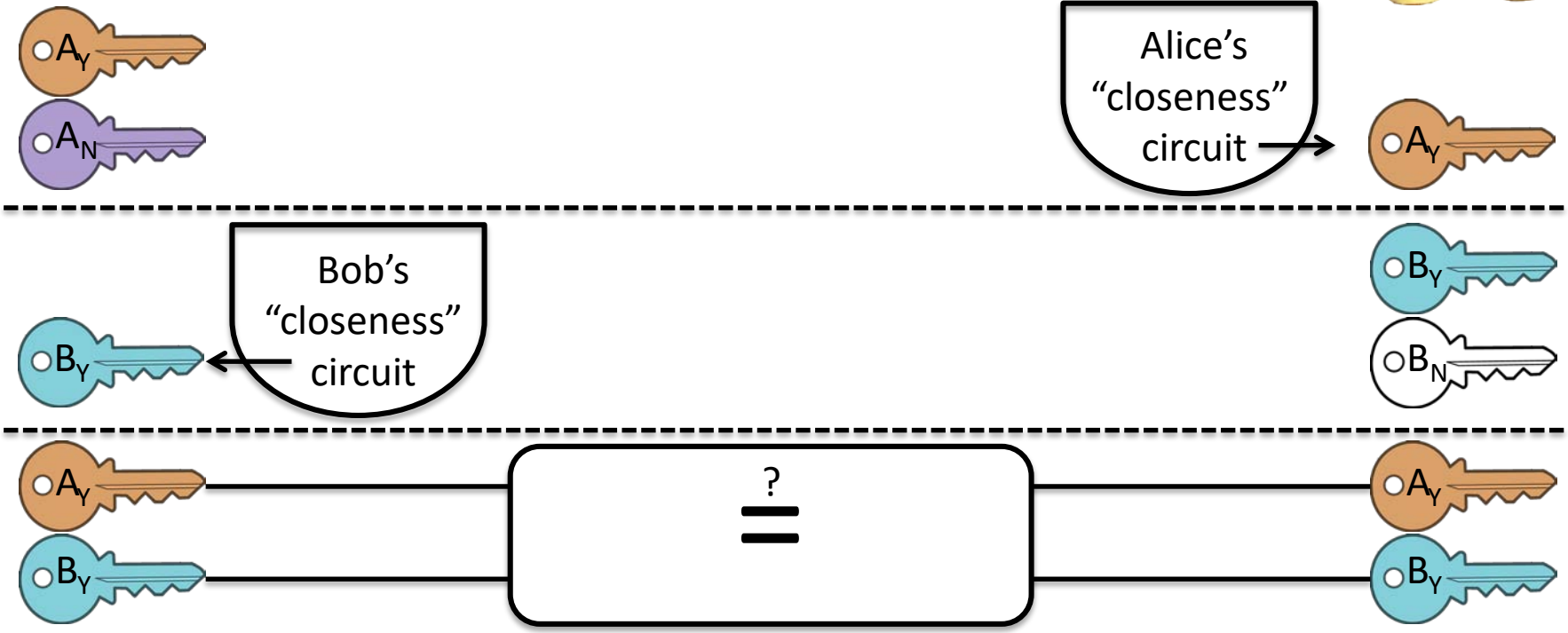
Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions



Do not reveal output to parties before comparison – always pretend that it is **yes**!

- Before: "Equal" => "computation correct", "Not equal" => "cheating"
- Now: "Equal" => "yes", "Not equal" => "no"

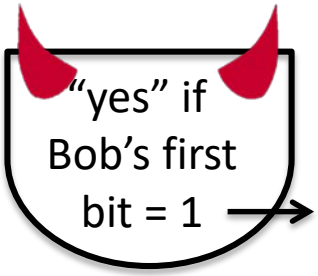
Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions

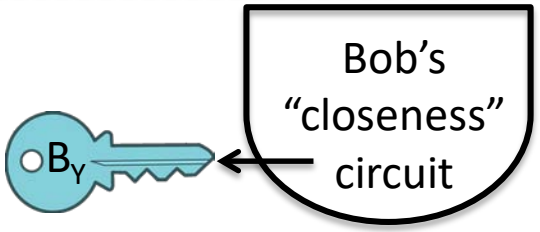


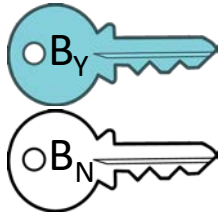
Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"		
	"no"		

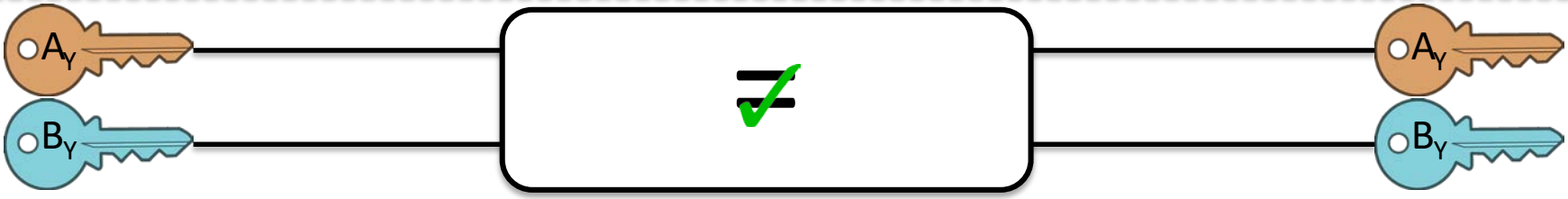
Do not reveal output to parties before comparison – always pretend that it is **yes**!

Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions





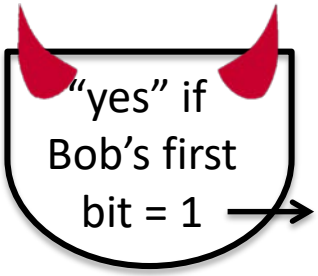


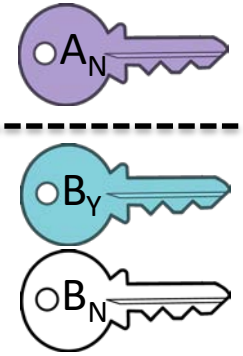


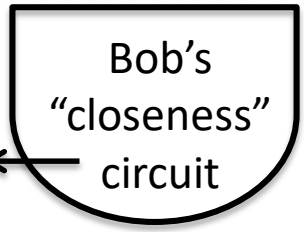
Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes"	1-bit leakage
	"no"		

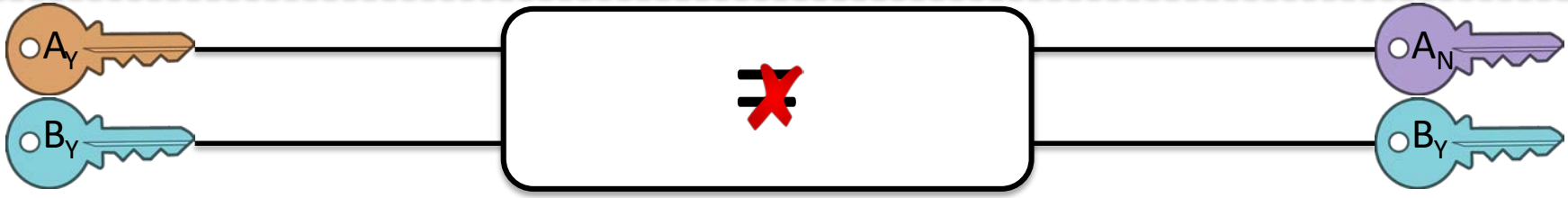
If Bob's first bit is 1, equality will succeed

Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions









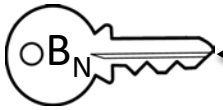
Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "no"	1-bit leakage
	"no"		

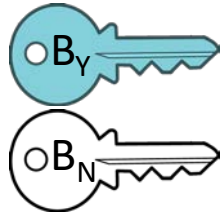
If Bob's first bit is 0, equality will fail

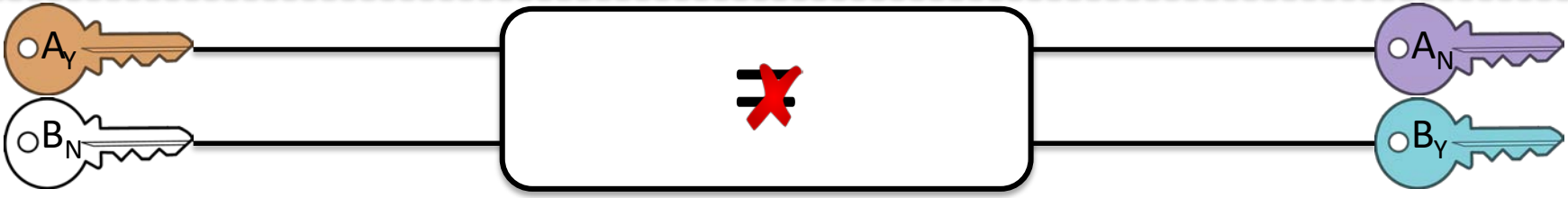
Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions

Alice's "closeness" circuit

Bob's "closeness" circuit







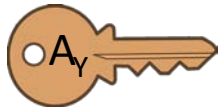
Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "no"	1-bit leakage
	"no"	"no"	

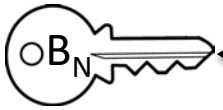
Do not reveal output to parties before comparison – always pretend that it is **yes**!

Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions

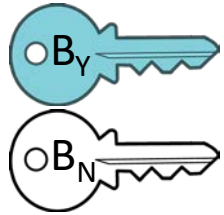
No clue about B_N !

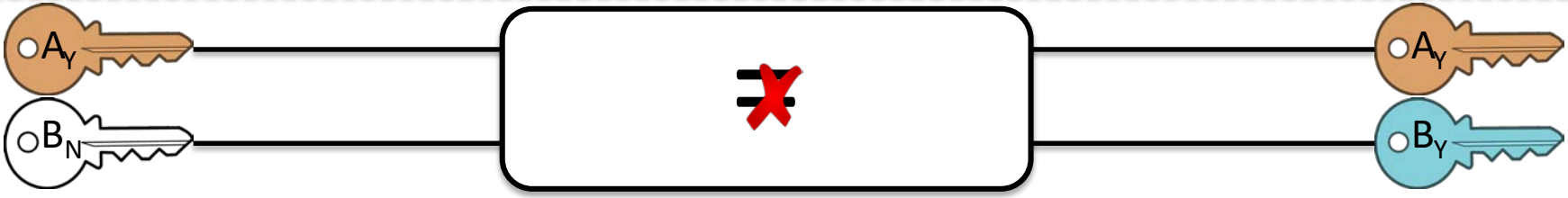
"yes" if Bob's first bit = 1





Bob's "closeness" circuit





Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "no"	1-bit leakage
	"no"	"no"	yes

Alice cannot learn anything additional – the output will be **no** no matter what she garbles.

Dual Execution for FPAKE: Privacy-Correctness Tradeoff for Boolean Functions

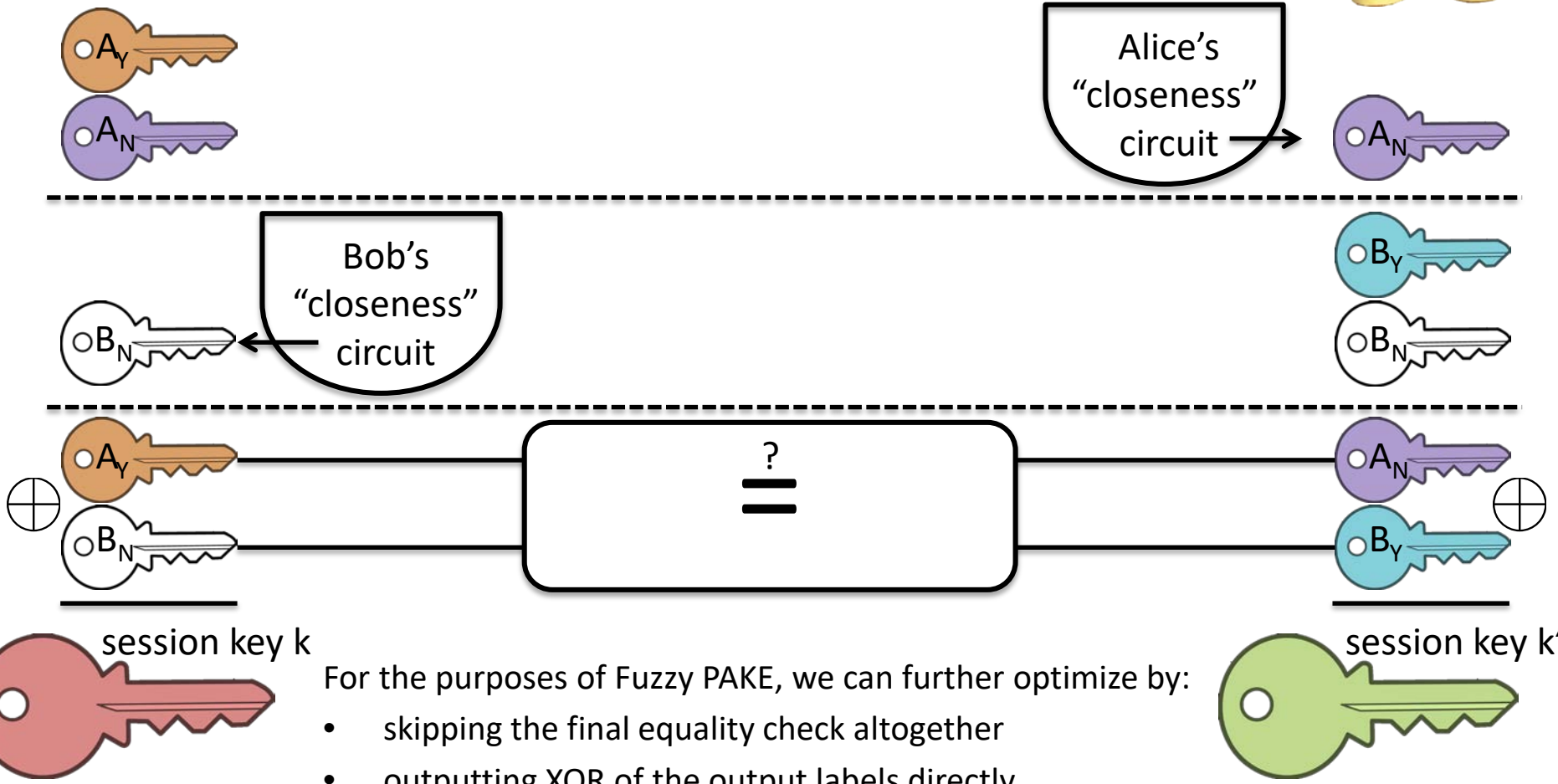
[MF'06, HKE'12] Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "cheating"	1-bit leakage
	"no"	"no" or "cheating"	1-bit leakage

Our FPAKE Dual Execution	Correct output	Comp. output	Privacy
	"yes"	"yes" or "no"	1-bit leakage
	"no"	"no"	yes

This is the perfect tradeoff for fuzzy PAKE!

- Only care about security against adversary who doesn't know a close-enough password – the "no" case

Dual Execution for FPAKE: Optimizations



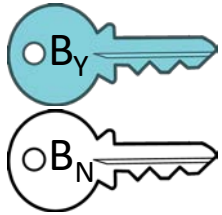
For the purposes of Fuzzy PAKE, we can further optimize by:

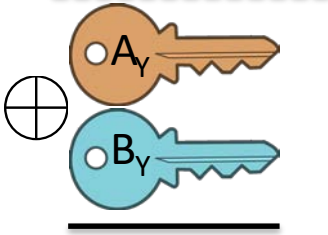
- skipping the final equality check altogether
- outputting XOR of the output labels directly

Dual Execution for FPAKE: Optimizations

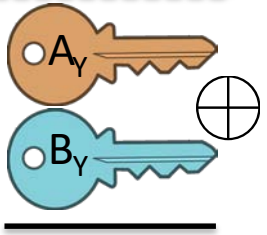
Alice's "closeness" circuit

Bob's "closeness" circuit





session key k



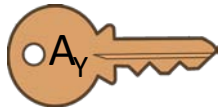
session key k

For the purposes of Fuzzy PAKE, we can further optimize by:

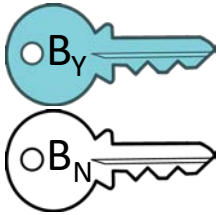
- skipping the final equality check altogether
- outputting XOR of the output labels directly

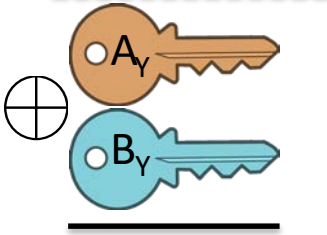
Dual Execution for FPAKE

Alice's "closeness" circuit



Bob's "closeness" circuit

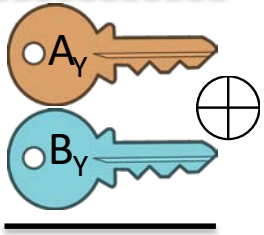




session key k

We're done!!

[BarakCanettiLindellPassRabin05]
+ YGC + modified dual execution
= our Fuzzy PAKE protocol!



session key k

Modified Dual Execution: More Generally

- Useful for functions where…
  - One output requires less security
  - The output is
    - Boolean, or
    - Same random / independent random
- E.g.:
  - Authentication
  - Mutual proofs of knowledge

Another Fuzzy PAKE Solution!

FPAKE construction	PAKE/Secret Sharing	Yao's Garbled Circuits
Notion of similarity	Hamming	Any
# rounds	2	5
# exponentiations	2n + constant	3n + constant

This talk

Conclusion

	Low-entropy password	High-entropy password
Exact match	PAKE	privacy amplification
Fuzzy match	New Primitive - Fuzzy PAKE	information reconciliation, robust fuzzy extractors

Our Contributions

- UC security definition of Fuzzy PAKE
- 2 efficient constructions
  - Including YGC with Modified Dual Execution