The Simulation Paradigm and Deniable Communications

Rosario Gennaro

Talk Outline

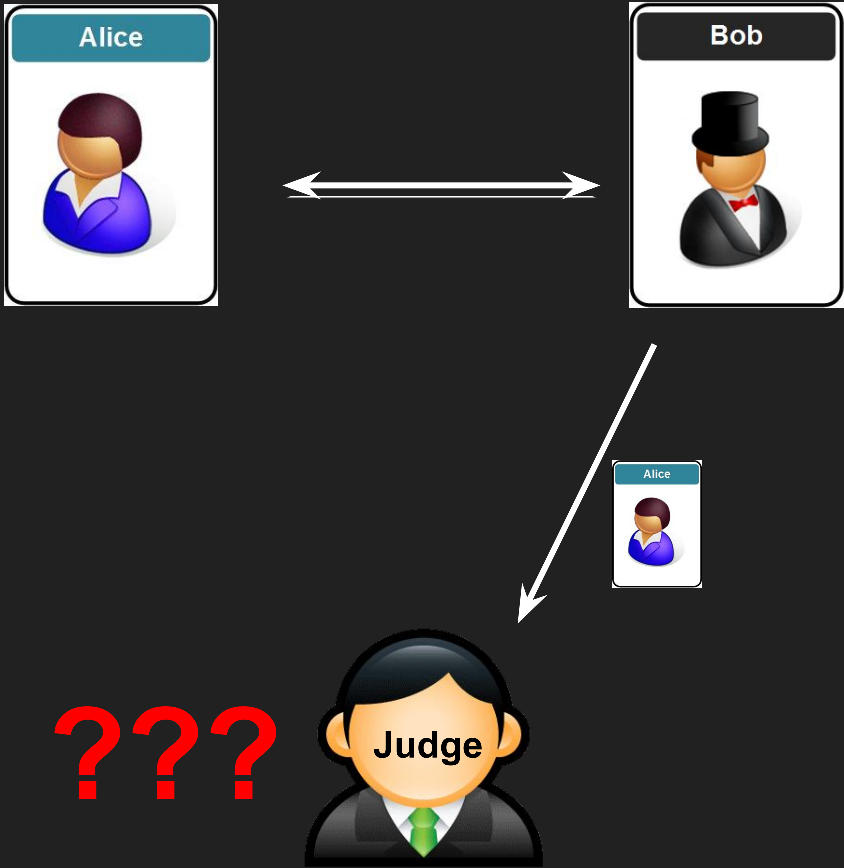
- Survey the notion of deniable or off-the-record electronic communication
  - Deniable Authentication [DDN]
  - Deniability as Simulation [Pass]
  - Deniable Authenticated Key Exchange [DGK]
- New results on the deniability of current internet messaging apps
  - This new work is in cooperation with my doctoral students Nihal Vatandas and Bertrand Ithurburn (CUNY) and Hugo Krawczyk (Algorand Foundation)

Deniable Communication

Two parties communicate

- They authenticate each other
    - *they verify each other's identity*

- They should not be able to prove that to a third party
    - *Off-the-record communication*



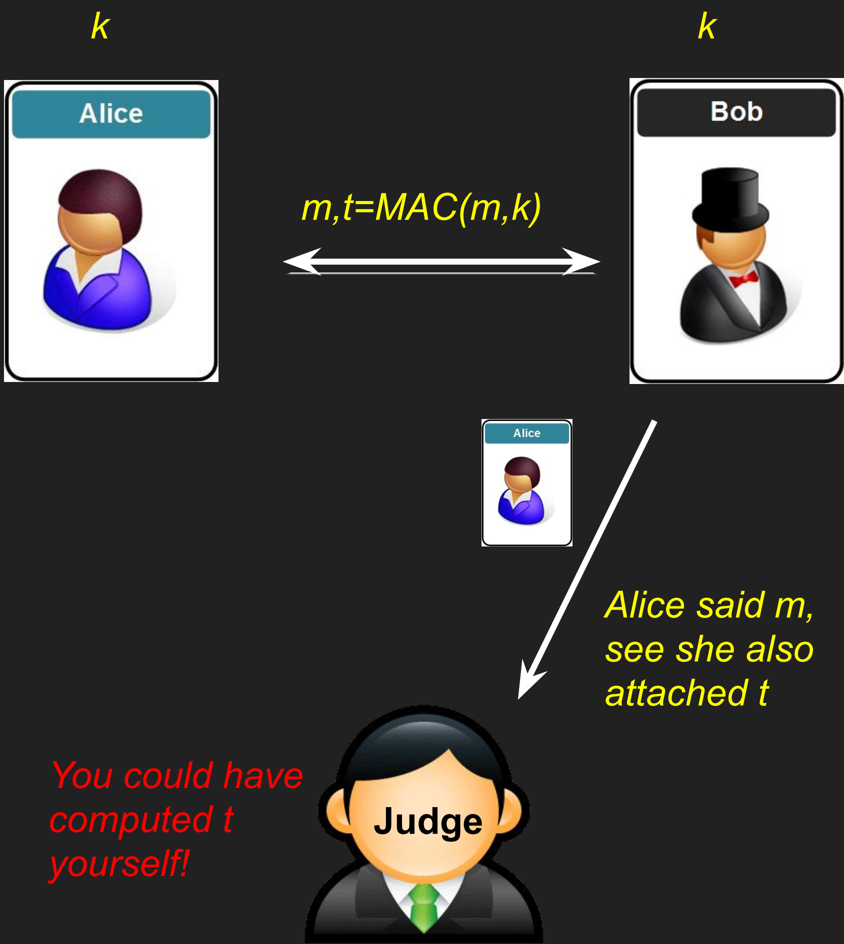
Cryptographic Authentication

Parties hold secret keys

- Attach to messages a function of the key that only the party can compute

Symmetric Key

- Both Alice and Bob hold the same key
- Messages authenticated by Alice could have also be authenticated by Bob
- Therefore deniable



k

k

m,t=MAC(m,k)

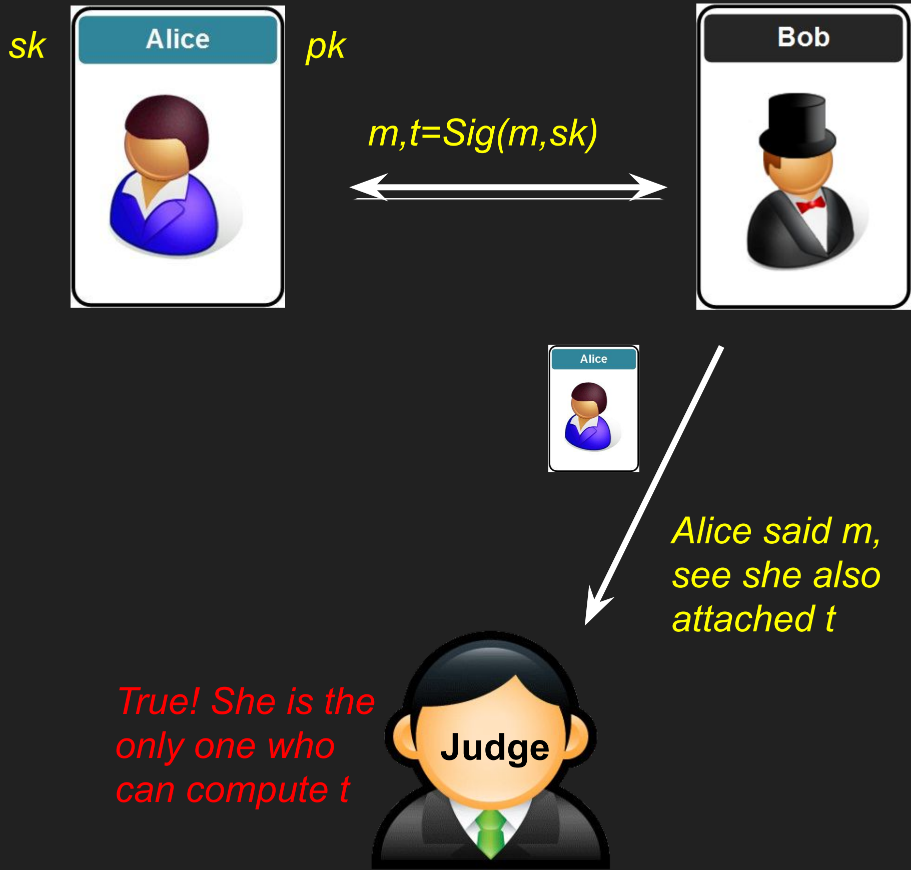
Alice said m, see she also attached t

You could have computed t yourself!

Cryptographic Authentication

Asymmetric Key

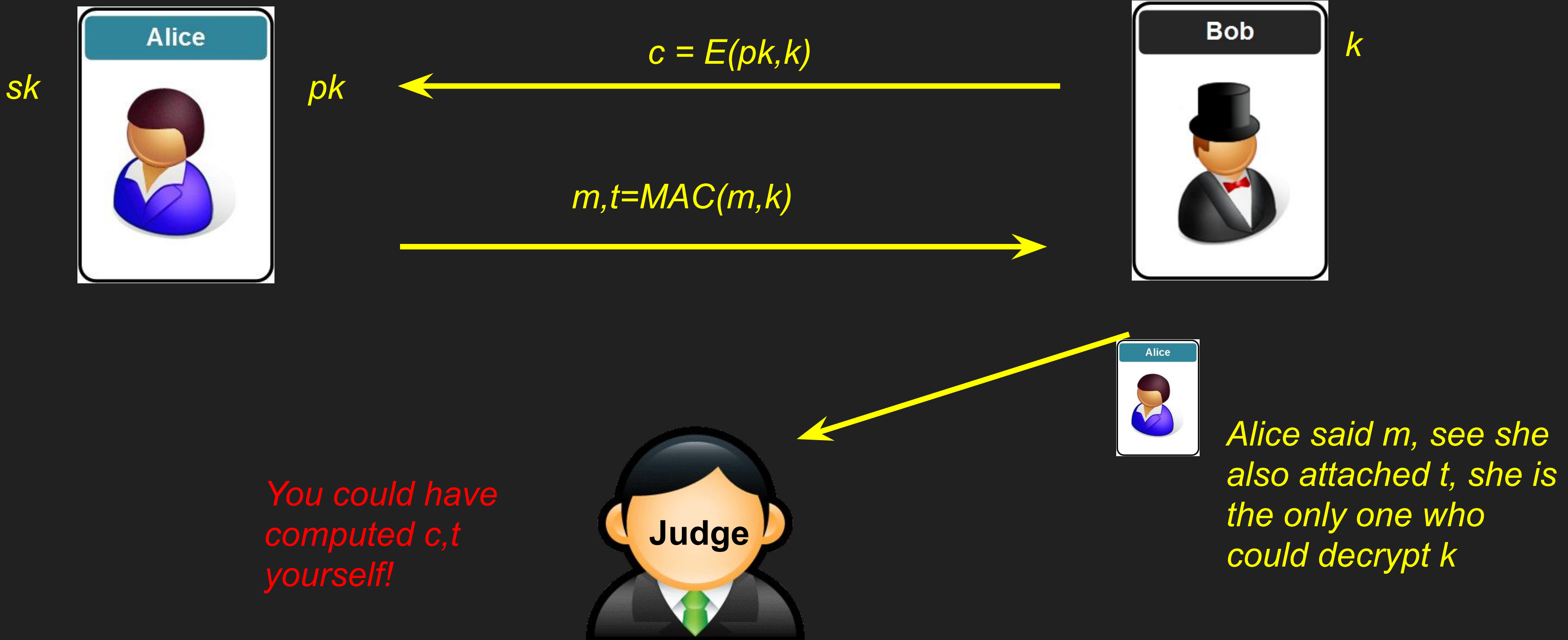
- Alice holds a secret key matching a public key associated to her
- She attaches a tag that only she can compute
  - e.g. a digital signature
- In general non-repudiable


sk
Alice
pk
m,t=Sig(m,sk)
Bob
Alice
Alice said m, see she also attached t
True! She is the only one who can compute t
Judge

Deniable Authentication

Asymmetric Key (DDN)

- Alice’s messages prove her identity only to Bob
- What Bob sees could have been produced by himself

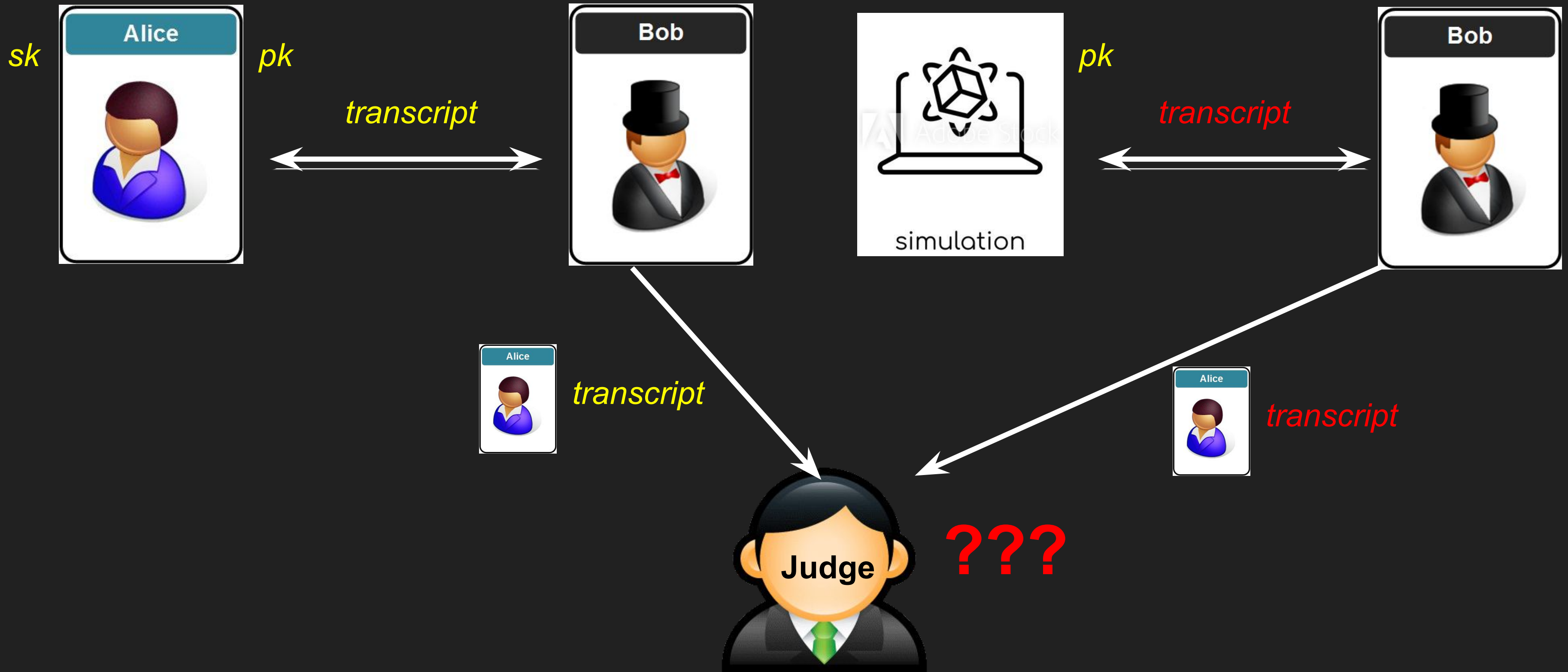


You could have computed c,t yourself!

Alice said m, see she also attached t, she is the only one who could decrypt k

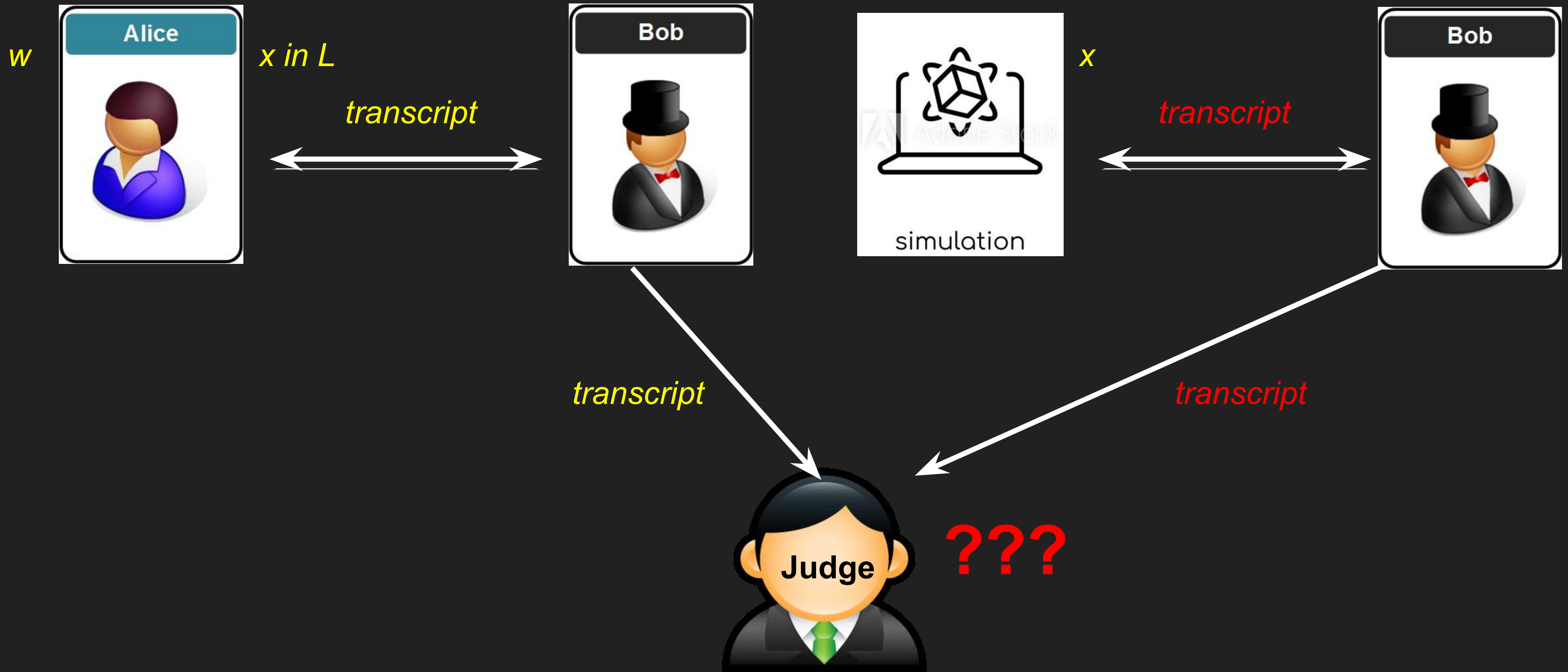
Deniable Authentication (DDN)

What Bob sees could have been produced by himself



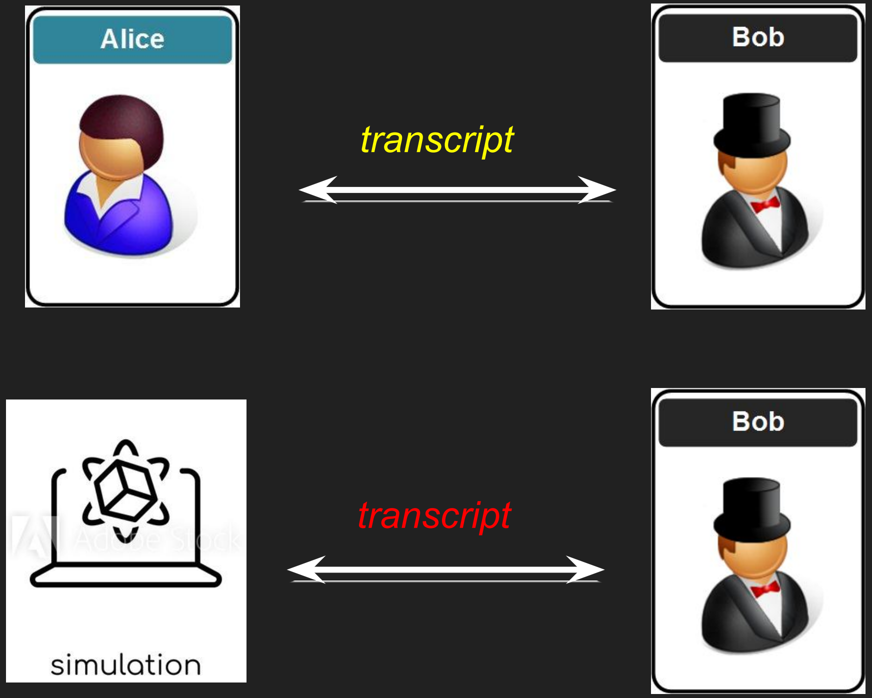
Zero-Knowledge (GMR)

What Bob sees could have been produced by himself



ZK Examples

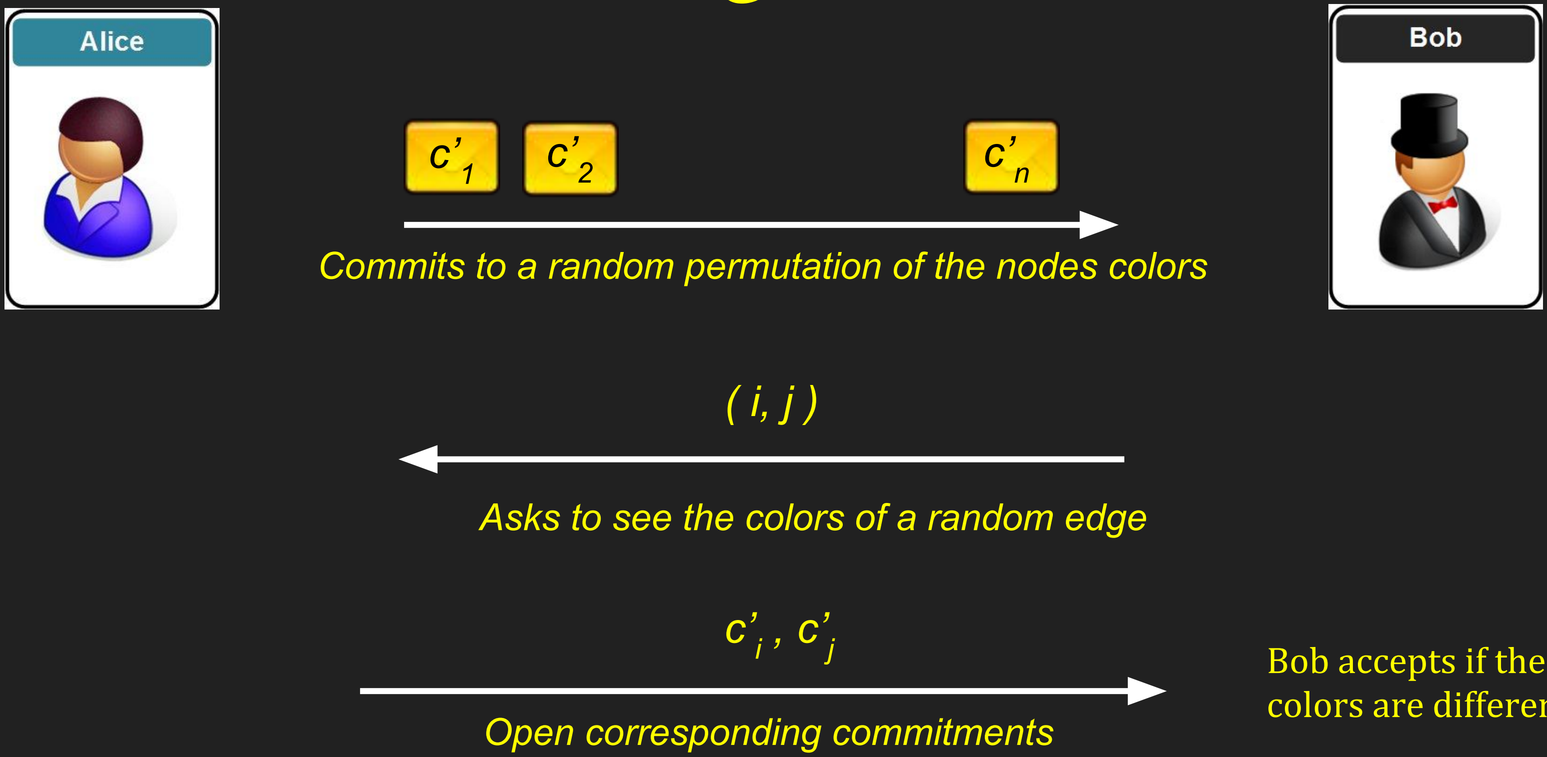
- A graph is 3-colorable
- Alice knows the factorization of a large number
- A Boolean formula is satisfiable
- Any NP problem can be proven in ZK



3-Colorability ZK Example

G

3-coloring:
$c(v_i) = c_i$



c'_1 c'_2 c'_n

Commits to a random permutation of the nodes colors

(i, j)

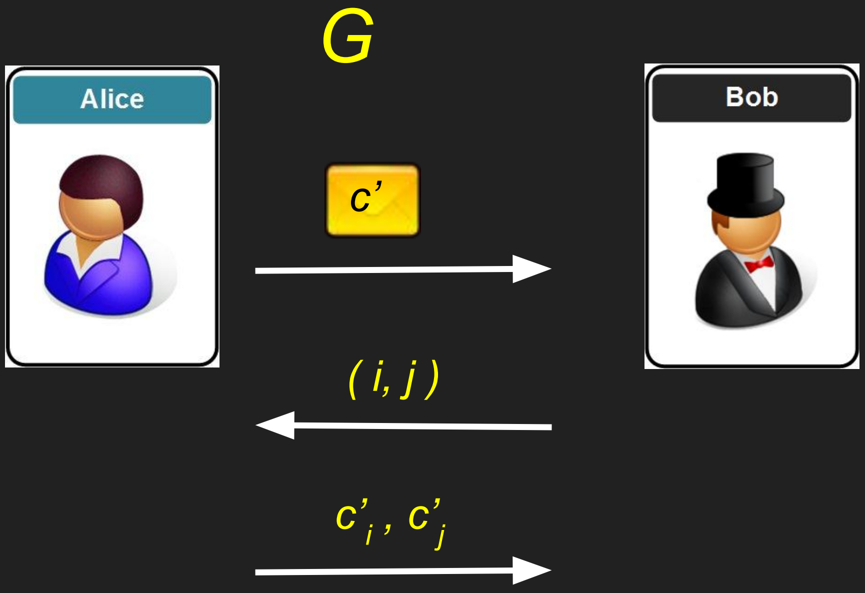
Asks to see the colors of a random edge

c'_i, c'_j

Open corresponding commitments

Bob accepts if the
colors are different

3-Colorability ZK Example



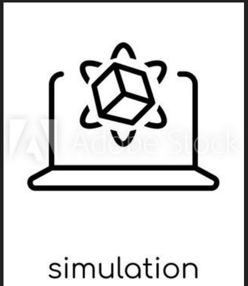
Why does it work ?

- If graph is not 3-colorable at least one edge must have same colors.
  - Probability *1/m* to catch Alice
  - Can be made smaller by repetition
- Bob only sees an edge with two different random colors
  - Colors are permuted for each repetition
  - Does not allow Bob to learn a 3-coloring of the graph

3-Colorability ZK Simulation

G

Does not know 3-coloring



c'_1 c'_2 c'_n →

Commits to a random coloring: at least one edge will be wrong

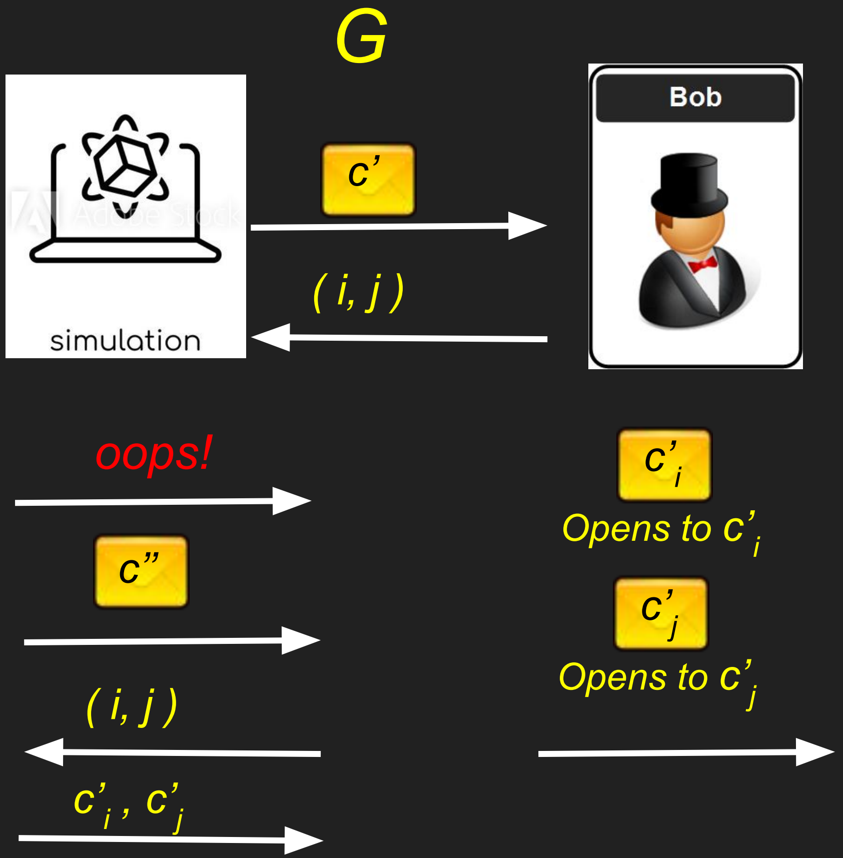
(i, j) ←

Asks to see the colors of a random edge

??? →

What if Bob asks the wrong edge?

3-Colorability ZK Simulation



Two classic simulation techniques

- Rewinding:
  - Simulator brings Bob back two steps
  - Change committed values so that requested edge is correct
- Random Oracle:
  - Commitment is done via a random function
  - Simulator is allowed to “program” the random function
  - Opens commitment at will to make it two different colors

Isn’t that cheating?

Short answer: Yes

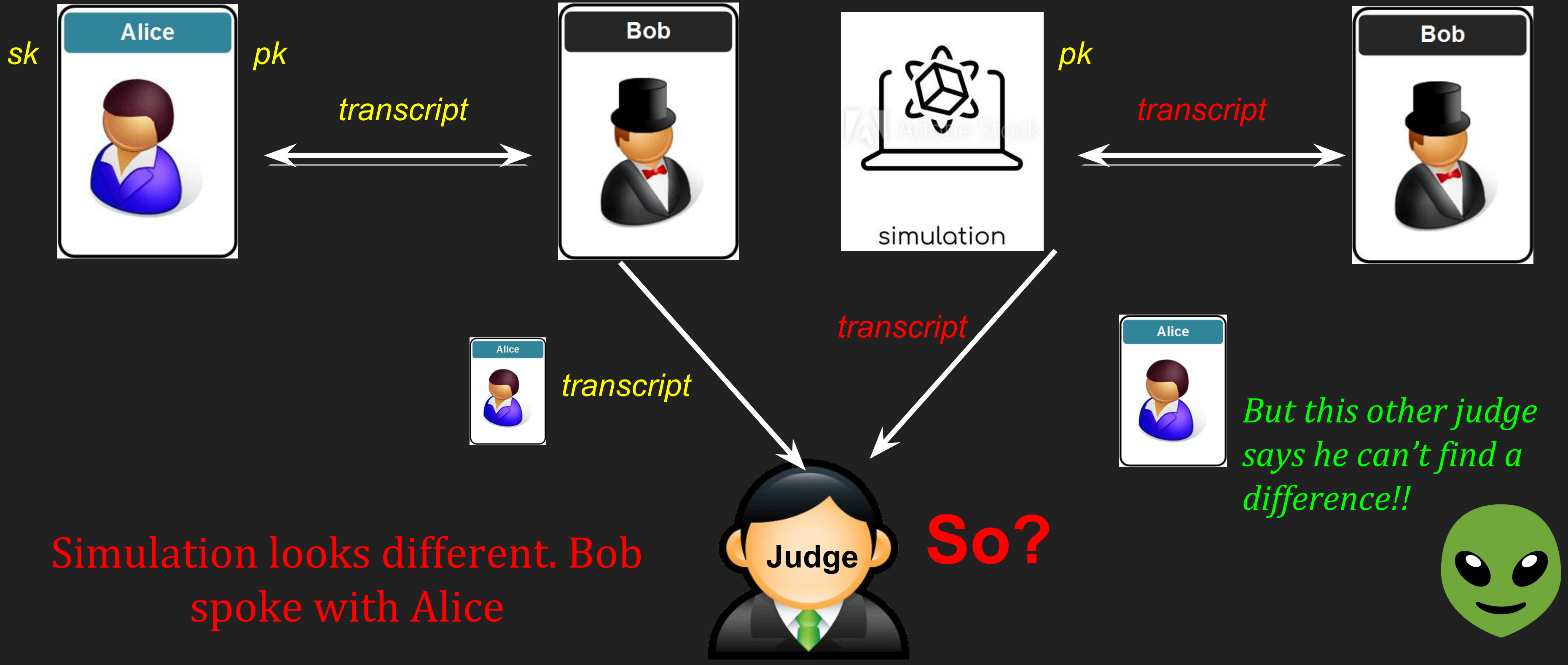
Long answer: it still proves that Bob learns nothing

- Simulator is a *thought experiment*

- We can set up a world where the conversation between Alice and Bob can be simulated without knowing any of the secrets of Alice
    - *Therefore the transcript itself contains no information about those secrets.*

- That’s where our judge, who decides if the transcripts look the same, lives

Is that OK for Deniability?



So?

But this other judge says he can't find a difference!!

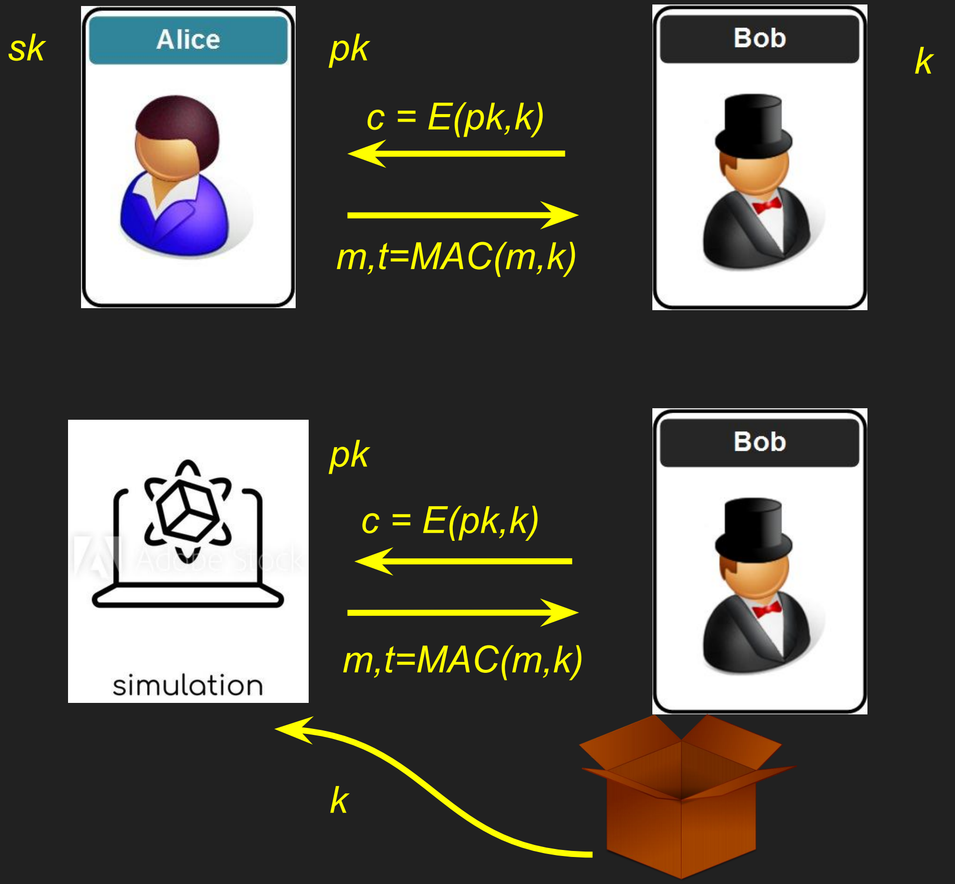
Simulation looks different. Bob spoke with Alice

Simulation for ZK vs Deniability

Pass 2003

- Deniability Simulation must work in the real world
  - Simulation must be straight-line
    - Rewinding is not allowed
  - Common Parameters are passed to the simulator and judge as input
    - The simulator is not allowed to choose them
    - Not allowed to "program" a random oracle
- Strong notion
  - Requires strong assumptions to be efficiently realized

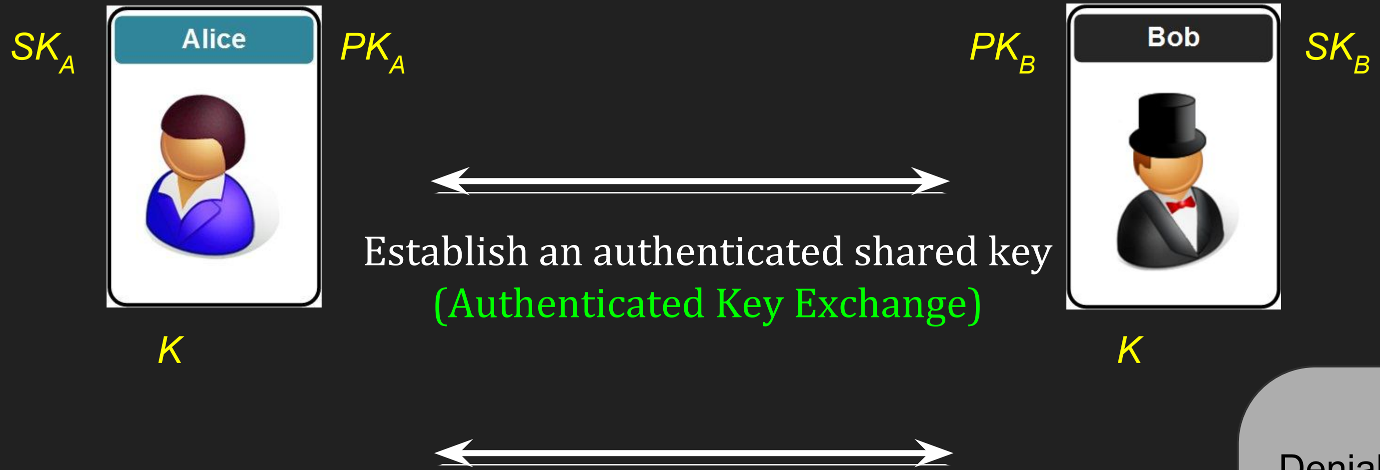
Encryption-Based Authentication



Simulator can’t decrypt ciphertext

- Must assume encryption is plaintext-aware
  - Valid ciphertexts can only be created if sender knows the corresponding plaintext
- Formally there is an extractor
  - When Bob outputs a ciphertext the extractor outputs the corresponding plaintext
- [DGK’06]

Communication Sessions



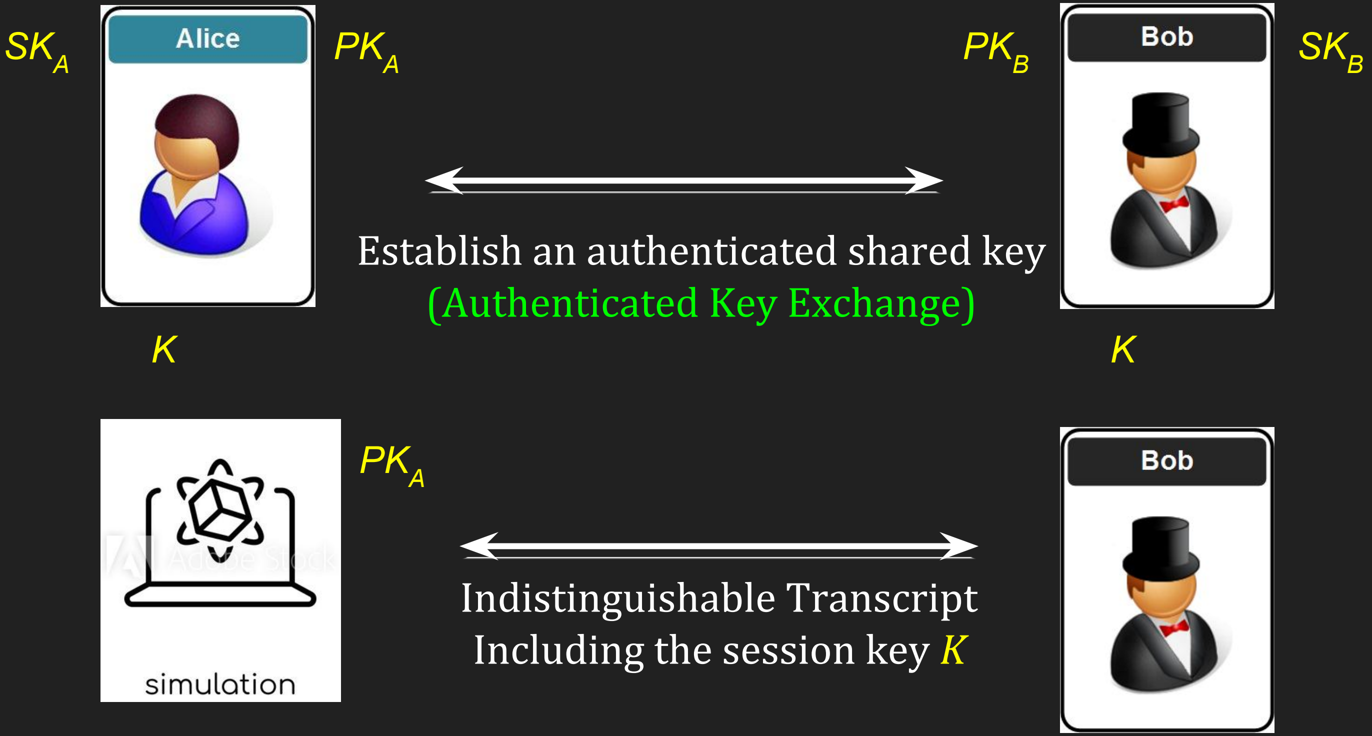
Establish an authenticated shared key
(Authenticated Key Exchange)

Protect communication using the shared key
(Secure Communication Session)

Deniable if a Judge believes that both Alice and Bob know the key K.

That requires the AKE to also be deniable.

Deniable AKE [DGK'06]



SK_A PK_A PK_B SK_B

Establish an authenticated shared key
(Authenticated Key Exchange)

K K

PK_A

Indistinguishable Transcript
Including the session key K

Deniable AKE [DGK’06]

- An Authenticated Key Exchange Protocol is deniable for Alice
  - If there exists a simulator running on input only Alice’s public key
    - Not her secret key
  - Simulator interacts with Bob (possibly malicious)
    - Real-world simulation
  - Creates a view that is indistinguishable from the real view
    - View must include the session key
    - Guarantees communication session is deniable no matter how the key is used

Deniability in AKE

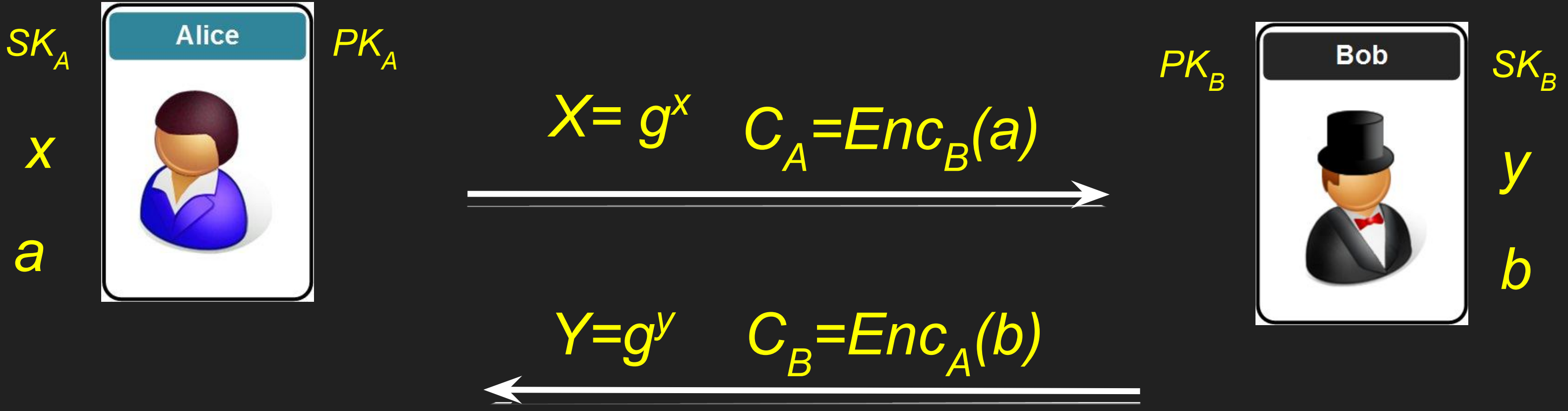
- Deniability was an important concern early on in the design of AKE
  - Informal discussions without a formal definition

- One of the first attempts to formalize and design deniable AKE was the influential Off-the-Record (OTR) protocol [BGB'04]

- Primary design consideration in new generation AKE protocols
  - Used in current messaging applications such as Signal, Telegram etc.

SKEME [K'96]

A Diffie-Hellman Key Exchange with encryption-based deniable authentication



$K'=Y^x=g^{xy}$

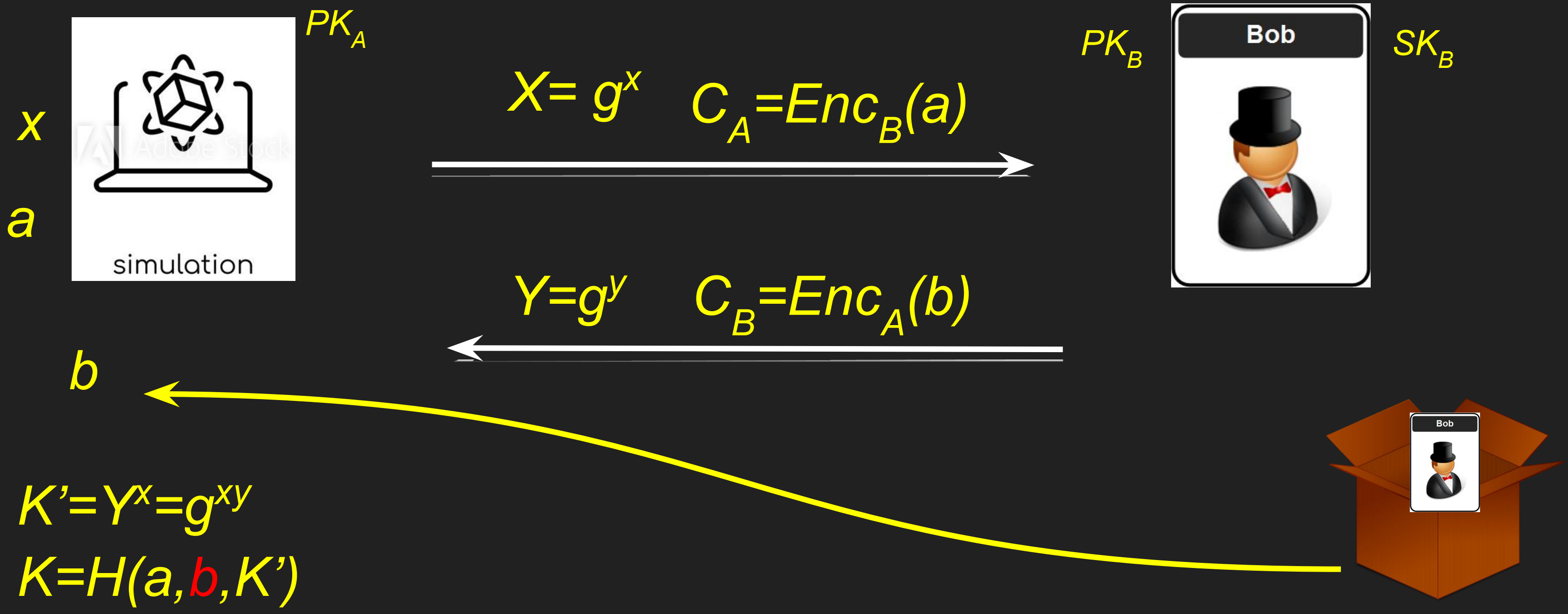
$K=H(a,b,K')$

$K'=X^y=g^{xy}$

$K=H(a,b,K')$

Simulation for SKEME [DGK’96]

Uses the plaintext-awareness of the encryption scheme



PK_A

x

a

simulation

$X = g^x \quad C_A = Enc_B(a)$

PK_B

Bob

SK_B

$Y = g^y \quad C_B = Enc_A(b)$

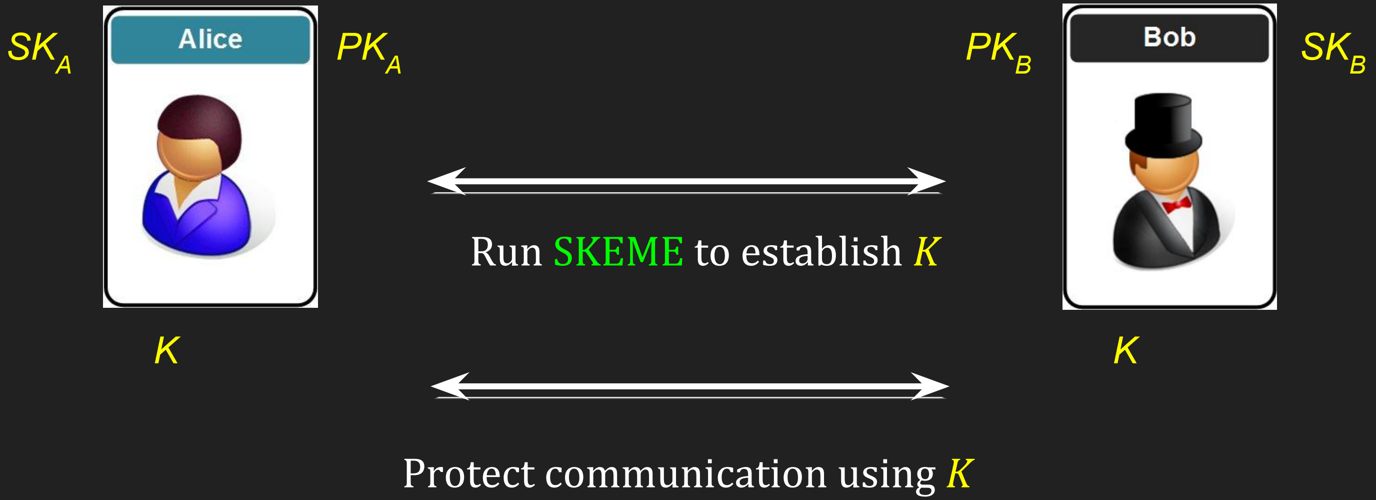
b

Bob

$K' = Y^x = g^{xy}$

$K = H(a, b, K')$

Deniable Communication Sessions



Run SKEME to establish K

Protect communication using K

We should be done, right?

Signal and 3DH

- New messaging protocols run a different AKE called 3DH
- Goal to avoid public-key encryption to authenticate
  - Potentially more expensive
  - Longer messages
  - Plaintext-awareness assumption
  - Bob's message depends on Alice's public key
    - Complicates an asynchronous mode in which Bob may not yet know with whom he is going to communicate
- But what type of deniable authentication is then used?
  - In spite of widely claimed and assumed deniability no formal analysis has appeared so far

3DH

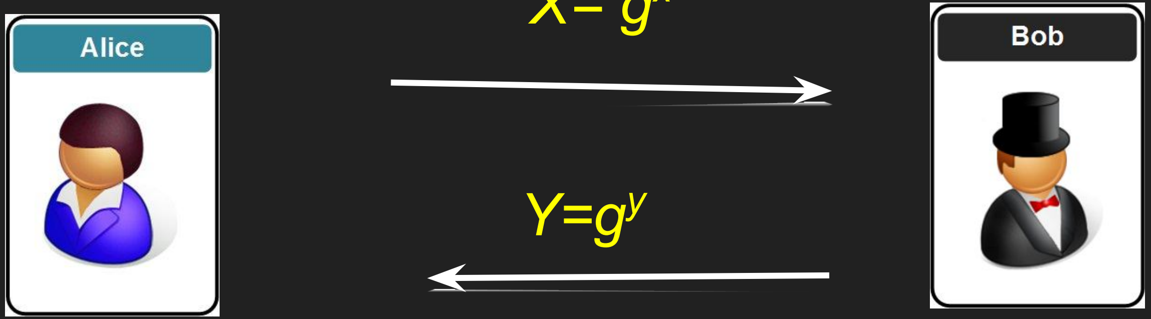
Triple Diffie-Hellman: A Diffie-Hellman Key Exchange authenticated via two additional Diffie-Hellman values.

$PK_A = A = g^a$

$PK_B = B = g^b$

x

a



$X = g^x$

$Y = g^y$

y

b

$K = H(Y^a, B^x, Y^x)$

$K = H(A^y, X^b, X^y)$

Simulation for 3DH?

$PK_A = A = g^a$

x

$X = g^x$

$Y = g^y$

$PK_B = B = g^b$

Bob could choose Y without knowing y

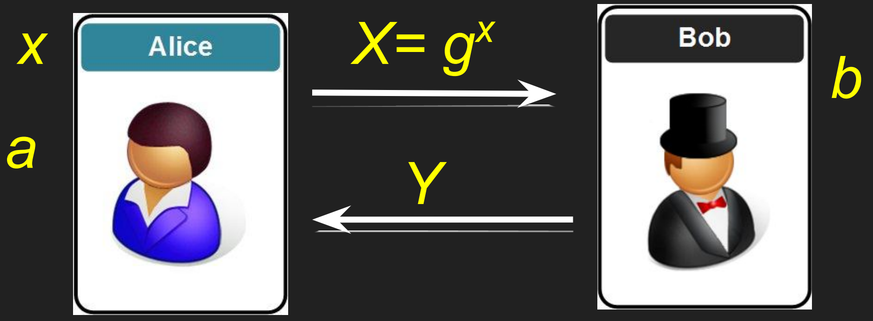
$K=H(Y^a,B^x,Y^x)$

y

Simulator does not know a or y

Simulation is impossible in general

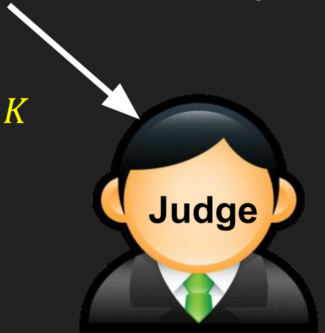
$PK_A = A = g^a$ $PK_B = B = g^b$



$K=Y^a.B^x.Y^x$

- Bob chooses Y so that he does not know y
  - E.g. hashing today’s newspaper
- Bob cannot compute g^{ay}
  - But this value is recognizable as correct
- In technical terms
  - Computational Diffie-Hellman is hard
  - Decisional Diffie-Hellman is easy
  - We know groups where this is the case

This is the correct K but I couldn’t compute it



True! Only Alice who knows a could do that!

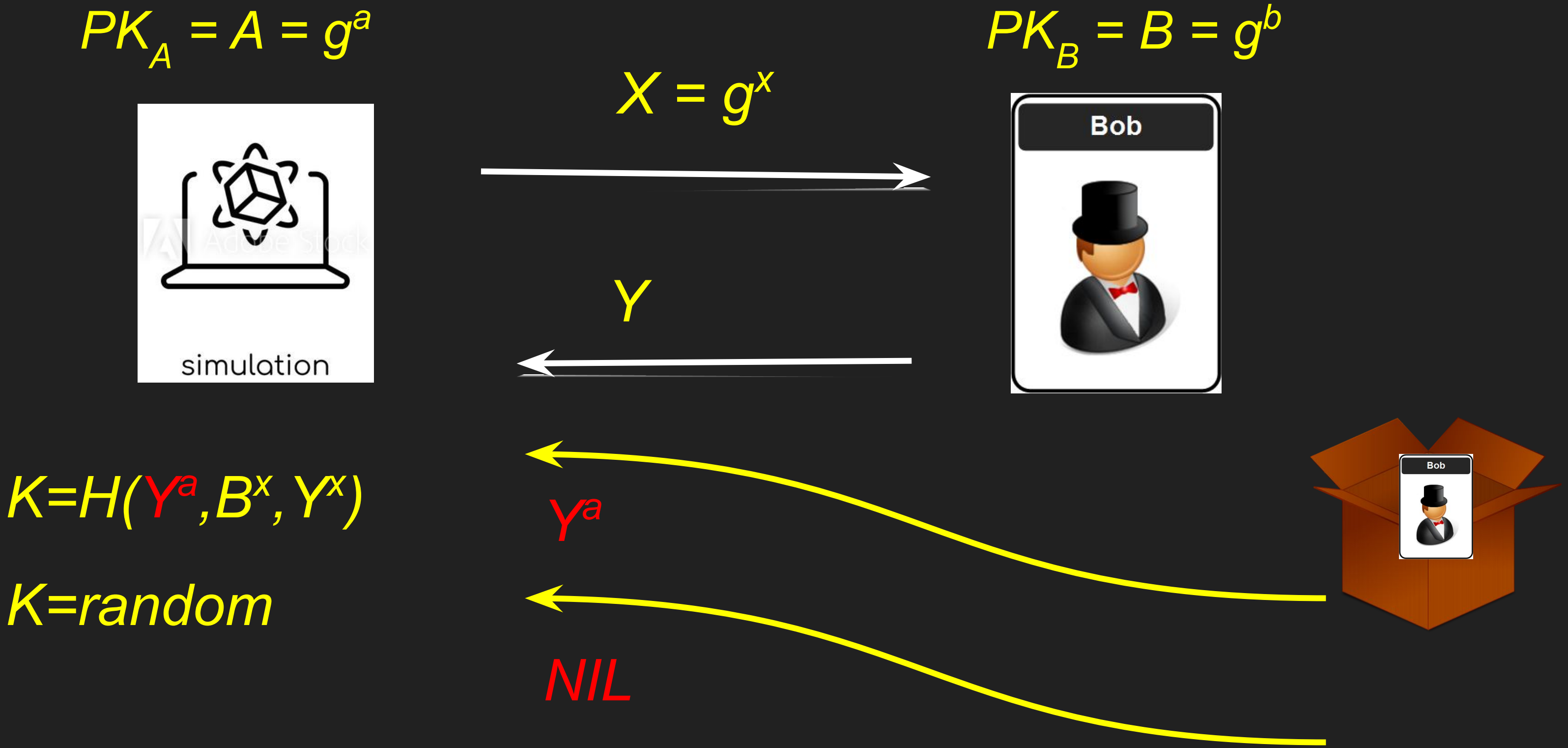
What’s the problem?

- In Signal, 3DH is implemented with $K=H(Y^a,B^x,Y^x)$ and in a group where DDH is assumed to be hard
- Still not sufficient to complete simulation since Y is adversarially sampled
  - We need to rule out a malicious sampling algorithm that
    - Chooses Y such that $H(Y^a,B^x,Y^x)$ is hard to compute but easy to detect as correct
    - Hardness of DDH does not help since Y is adversarially sampled
    - Modeling H as a random oracle does not help either as we cannot detect the correct query to find Y^a

Simulation for 3DH

Simulator does not know a or y

x



$PK_A = A = g^a$

$PK_B = B = g^b$

$X = g^x$

Y

$K=H(Y^a,B^x,Y^x)$

Y^a

$K=random$

NIL

Simulation for 3DH

- We assume the existence of an extractor such that
  - When Bob outputs Y
  - The extractor outputs either Y^a
    - This means Bob knows Y^a and the protocol is deniable since that means there is no proof he got it from Alice
  - Or the extractor outputs NIL
    - This means that nobody can distinguish the key K from a random value and therefore again what Bob presents to the judge is meaningless
- This is a strong assumption
  - Related to the *Knowledge of Exponent Assumption*
    - We can reasonably assume that it holds for the groups used by Signal

Alternatives to 3DH

- Signal allows X3DH
  - Asynchronous version of 3DH

- Bob loads Y on a server when he goes offline
  - This allows anybody to send him a message while he is offline
    - Read Y from the server
    - Run 3DH to compute K
    - Encyrypt/Authenticate the message with K and leave it with the server for Bob

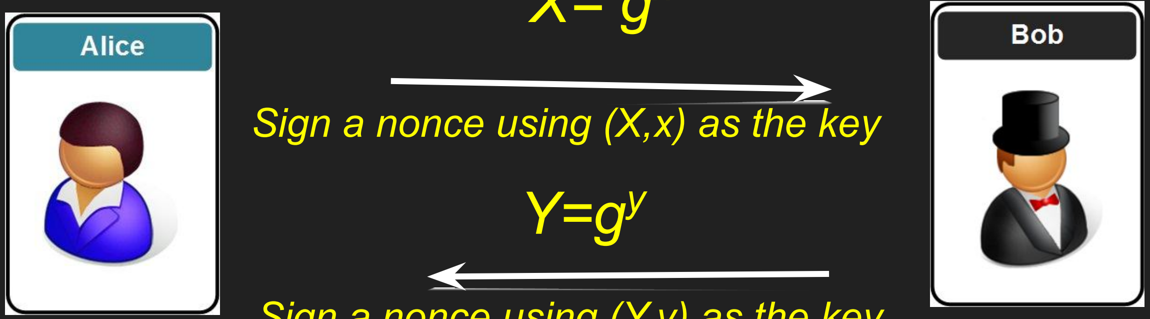
- Important Property:
  - Bob's AKE message cannot depend on the identity of the party he will communicate with
    - Rules out SKEME

Alternatives to 3DH

Prove knowledge of *y*, via a signature

$PK_A = A = g^a$

$PK_B = B = g^b$



x

a

$X = g^x$

Sign a nonce using (X,x) as the key

$Y = g^y$

Sign a nonce using (Y,y) as the key

y

b

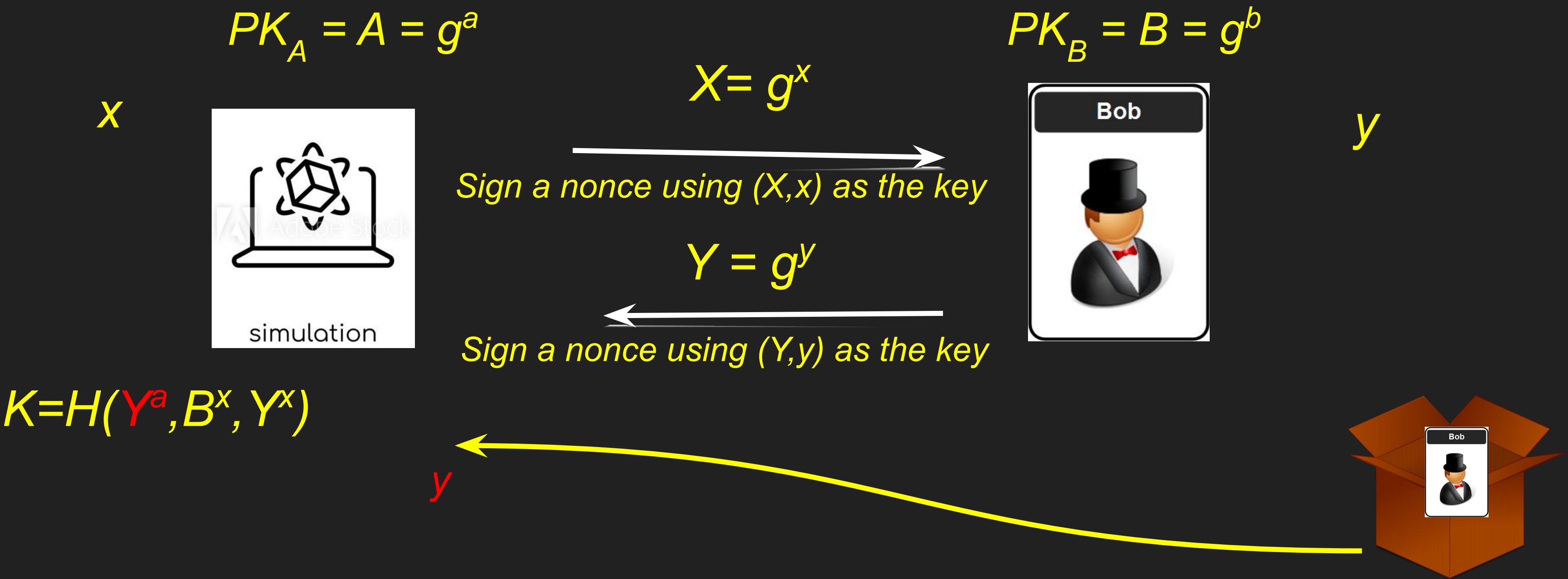
$K = H(Y^a, B^x, Y^x)$

$K = H(A^y, X^b, X^y)$

Prototype implemented by undergraduates at CCNY (J.Moore, K.Natavio, N.Rea, A.Timashova)

Simulation for 3DH-Alt

Prove knowledge of *y*, via a signature



$PK_A = A = g^a$

x

$X = g^x$

Sign a nonce using (X,x) as the key

$Y = g^y$

Sign a nonce using (Y,y) as the key

$PK_B = B = g^b$

y

$K=H(Y^a,B^x,Y^x)$

y

Conclusions

- Deniable off-the-record communication is crucial to allow truly anonymous interaction
  - Important societal implications: whistleblowers, human right activism, journalism, etc

- There is a mature body of research that formally defines what this means
  - Problem is hard and require strong assumptions on the hardness of certain computational tasks
  - The stronger the assumption the least confidence we have that it holds
  - We need to keep looking for solutions with the weakest and most reasonable assumption

- Trust but verify
  - Protocols may seem intuitively deniable
  - Proving their deniability is still necessary

Conclusions

- Simulation is a tool.
    - When we standardize ZK and Simulation we need to keep in mind what the application is
        - Deniability simulation is a different beast than ZK simulation

- Protocols chosen for standardization should be thoroughly vetted and formally proven
    - Signal is an impressive piece of work and was a game-changer in the area of internet messaging apps
    - Yet we should not have let years pass without a formal analysis and proof of one of its most crucial features
        - See [ACD’19] for belated and needed formal analysis of other security features of the Signal protocol