# The Multiple Faces of Deniability

Rafail Ostrovsky
UCLA

*NIST Special Topics on Privacy and Public Auditability — Event 7; January 16, 2025*

# Ideal Channel (or shared 1-time pad)

- A and B have 1-time pad R
- They send messages to each other by XOR'ing these with 1-time pad
- They can both later deny what was said by claiming that R was different.
- They can both lie; it is impossible to determine who lies and who tells the truth.
- Quantum Key-Distribution (if used as a 1-time pad) is fully deniable

- Q: can we do it for public key encryption instead of 1-time pad?

# Reminder: PKE

- Key-gen($1^k$,r) → (PK,SK)
- E(m,PK,r) → c
- D(c,SK) → m

- Semantic security: Alice sends c to B, where c is either m1 or m2

- Eve hears c, but cannot tell if c encrypts m1 or m2

# The setting

Alice → c → Bob

Eve hears c, comes to Alice, and asks her what m was encrypted inside c?

(Eve asks Alice to show her the message and randomness that produced c ….or else….)

Not an issue for the “ideal” private channel.

Alice has several options:

“I forgot what I said” (not always possible), or

use “deniable encryption”

# What is deniable encryption?

- Sender deniable (Receiver understands clearly what the message is, but Sender can come up with fake message for the same c)

- Other deniability notions:
- Receiver Deniable
- Sender and Receiver Deniability (by-deniable).

# A simple trick

- Assume trapdoor permutation family $(f, f^{-1})$
- Given a seed s of k bits, and using f to construct Pseudo-Random Generation $PRG(s)=y_1,\ldots\ldots y_{3k}$
- Using $f^{-1}$ we can efficiently *test* if the output is random or pseudo-random

Alice: to send a 0, just send a random string of length 3k bits
to send a 1 send an output of PRG

Alice can now ``lie'' on 1, (by converting 1 to 0) (but not the other way around)

To amplify: use XOR of many bits to encode a single bit… can get 1/poly deniability

Deniable file system, Julian Assange, using "shaff layers".

# Sender deniability

- Consider court order to preserve documents, you cannot say, “I forgot”, but if you encrypt using  Sender-deniable encryption, you could decrypt to anything, and in fact, you cannot even later prove that what you decrypted is correct, even if you want to collaborate with the investigation…

- What are other non-nefarious applications?
  - Preventing coercion in voting (sort of…)

# From sender deniability to receiver deniability  R

- Sender and Receiver:
- First, R sends deniable encryption bit b to S
- S sends received bit b XOR “her input bit”.  Now, R can deny what it received by denying what it sent in the first round.

- Deniability for both?
- One possible answer: User threshold with a bunch of intermediaries
- Another answer: use obfuscation…

# State of the art

- Canetti-Dwork-Naor-Ostrovsky 96
  - (based on trapdoor permutations)
  - Sender-deniable (with 1/poly deniability)
  - Receiver-deniable (with 1/poly deniability)
- Bendlin, Nielsen, Nordholt, and Orlandi 2011: any receiver-deniability must take at least 3 rounds of interaction
- Sahai-Waters 2014
  - Use obfuscation to construct both sender and receiver-deniable encryption
    - 2-rounds, 3-rounds respectively
- Canetti,Park, Poburinnaya 2019
  - Use obfuscation to construct bi-deniable encryption in 3 rounds (& no “incriminating”)
- Coladangelo, Goldwasser, Vazirani 2022
  - Sender-deniable Quantum Encryption with classic cyphertext/decryption

# A related notion: Non-committing encryption [CFGN 96]

- Receiver has only 1 (out of 2) decryption keys.
- Even with one decryption key given to R (out of two), R understands encrypted msgs without error for any of the two keys
- Can “maliciously” encrypt, so that the meaning of encryption “toggles” as a function of which of the two encryption R has.

- Who cares?  Inside CRYPTO proofs, can simulate “ideal” channel
- Yes, but still who cares? Can prove composability…

# Equivocate Commitment

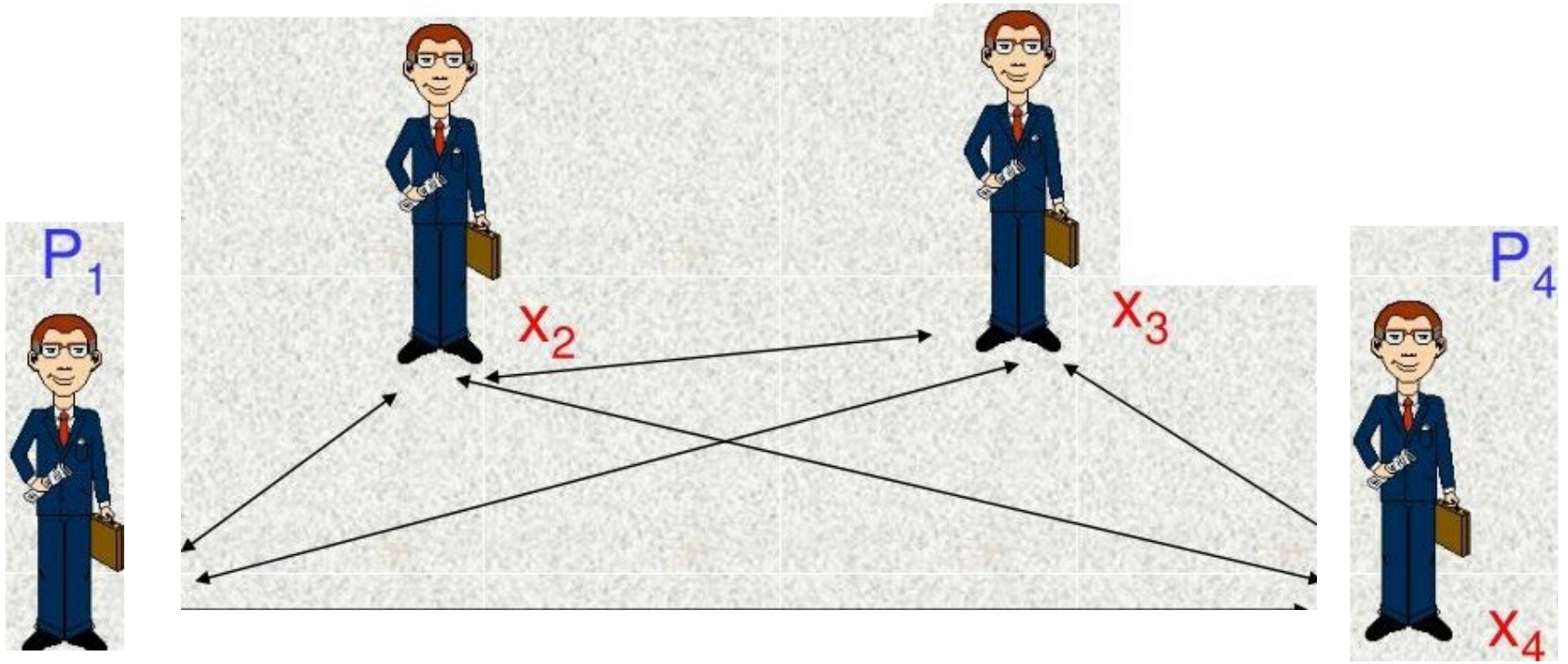
- Standard commitment is binding.
- We can make it non-binding with a special “trapdoor” that could be generated by the ``simulator’’ but not by the participating parties
- Who cares?  Key tool defending MiM attacks in protocols:

A ←→ Men-in-the-middle ←→ Receiver

# Somewhere equivocal commitment [HJOSW 16]

- Used inside Adaptive Garbled circuit constructions
- Function Seccret Sharing: [BGI]; Can generate s1 and s2 s.t.
  - PRG(s1) xor PRG(s2) is a unit vector, but separately s1 or s2 do not reveal what j  unit vectoreis one.
  - This can be used to commit (xor with PRG output) where in a secret place can toggle between s1 an s2.
  - Can be used for *Adaptive Securit*y of Garbled Yao and Garbled RAM
    - **(secure computation as a service)**

# Covert Multi-party Computation [CGOS 07]




$f(x_1,x_2,x_3,x_4)$

No information other than $f(x_1,x_2,x_3,x_4)$

# Do all of us want to rebel??


Powerful Dictator

# Crypto Solution [Yao,GMW]

Rebel= 1

No Action= O

$P_1$ 1

Multi-party Computation AND(inputs)

0

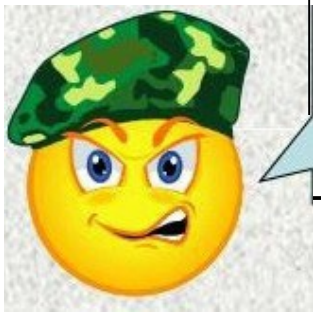
1

$P_3$

# INFORMAL PROBLEM STATEMENT

Introduced by [von Ahn, Hopper, Langford '05]

# Crypto Solution [Yao,GMW]



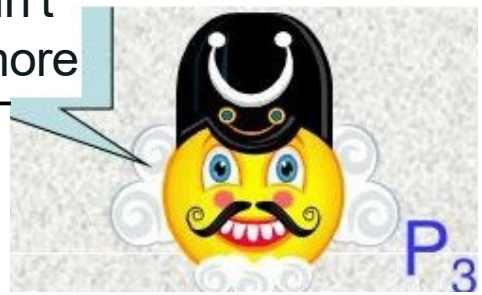
Lets run MPC to see if all of us want to rebel


**$P_1$ wants to rebel!!**

# Ideally

How are you guys?

I couldn't agree more




All of us want to rebel!!

Doing well.. Army life is hectic...

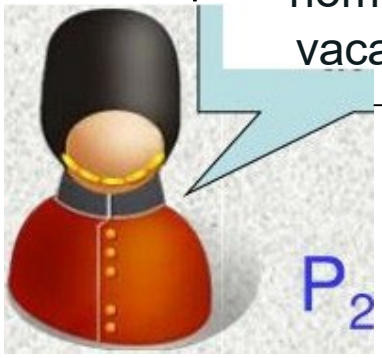
# Ideally

How are you guys?

Oh.. That's fantastic!


Someone does not want to rebel or did not participate!!

Not too bad.. I am going back home on vacation



# Covert Computation

Can we reveal protocol participation and output only upon the condition that everyone participated and output was favorable ??

- Two party case (restricted): [AHL05]
- Multi-party case: [CGOS07]

# Privately Updatable Encrypted File System
## (Hiding Which Files are Updated)

- This is well studied under the name of Oblivious RAM (ORAM)
  - [Goldreich Ostrovsky 96]
  - By now, many fast solutions, thousands of papers
  - Who cares? (story time)

  Two models

  Client-Server (small client, cloud)

  Multi-Server MPC setting which supports CPU—Memory reads/writes

# Conclusions

Deniability is everywhere in CRYPTO. Some examples:

1. ZK: ``faking'' conversation with the prover, without knowing the witness
2. UC: must simulate protocols without even knowing what the outcome of the conversation is
3. Adaptive security (having to do with the power of th e adversary)
4. Lossy Encryption and Perfect NIZK: two modes of encryption
5. Circular-secure encryptions [BHHO 08]
6. Leakage-resilient Encryption [NS 09]
7. Non-interactive Secure Computing
    1. Alice generates PK, which somehow embeds f(*,y), for poly-time two-argument function f
    2. Bob with x, encrypts E(x), sends to Alice, and Alice learns f(x,y) and nothing else

Many more…

# THANK YOU!