Efficient and Fair MPC using Blockchain and Trusted Hardware

Souradyuti Paul
(IIT Bhilai)

Ananya Shrivastava
(IIT Gandhinagar)

Latincrypt 2019 Santiago, Chile
October 3, 2019

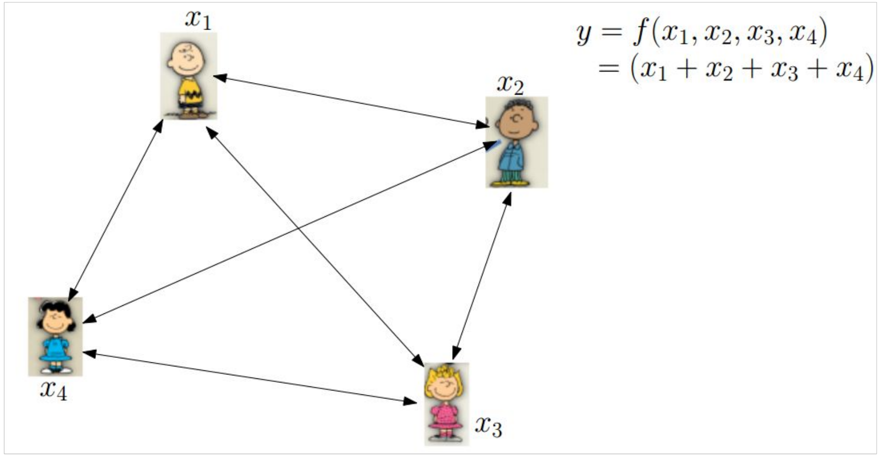
Outline

- ❑ Multiparty Computation (MPC)
  - ❑ Security Property of MPC: Privacy, Correctness, Fairness
- ❑ Various Components
  - ❑ Blockchain
  - ❑ Trusted Hardware
  - ❑ Core MPC having *privacy* and *correctness* security
- ❑ Fair MPC Protocol using Blockchain and Trusted Hardware: CGJ+ Protocol
- ❑ Attack on CGJ+ Protocol
- ❑ Our Construction
- ❑ Results

Multiparty Computation (MPC)

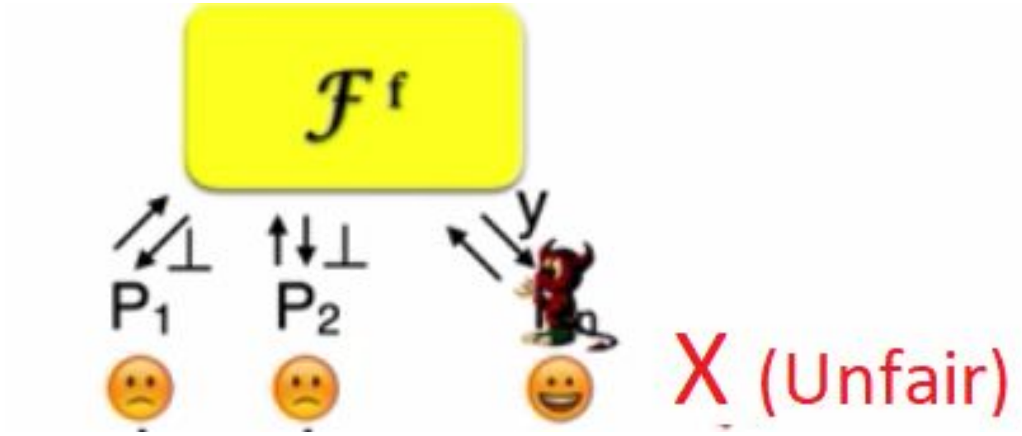
Definition (Informal)

There are *n* parties P_1, P_2, …. , P_n who do not trust each other. Each party P_i has its own private input x_i and there is a common function *f*(.) with *n*-bit input that every party wants to compute on their private data.



Security Property of MPC: Fairness

Definition (Informal)
An adversary can receive their output only if all honest parties receive output.

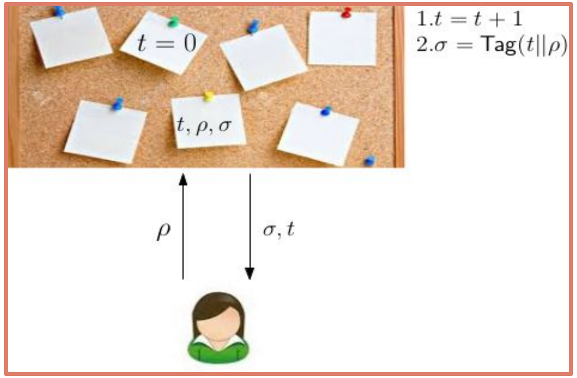


Component 1: Bulletin Board (Blockchain)

Properties:

- Messages are permanently available.
- Messages are visible publicly to all the parties.
- Produces a publicly verifiable proof that the message is posted publicly.
- Generates proofs using an *Authentication Scheme* which can be publicly verified.

Public Ledger BB

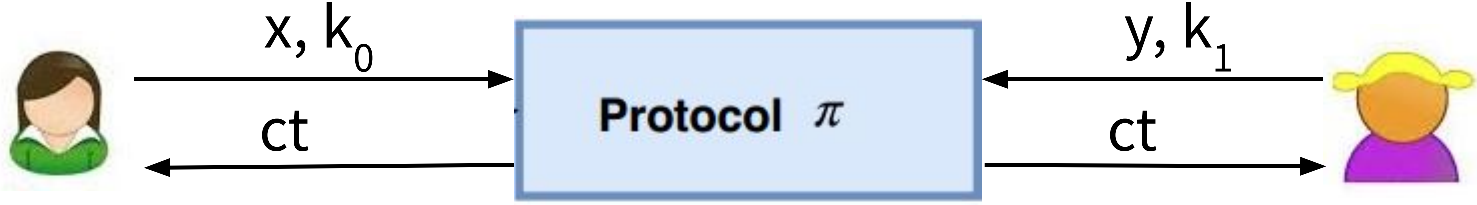


Component 2: Trusted Hardware

Properties:

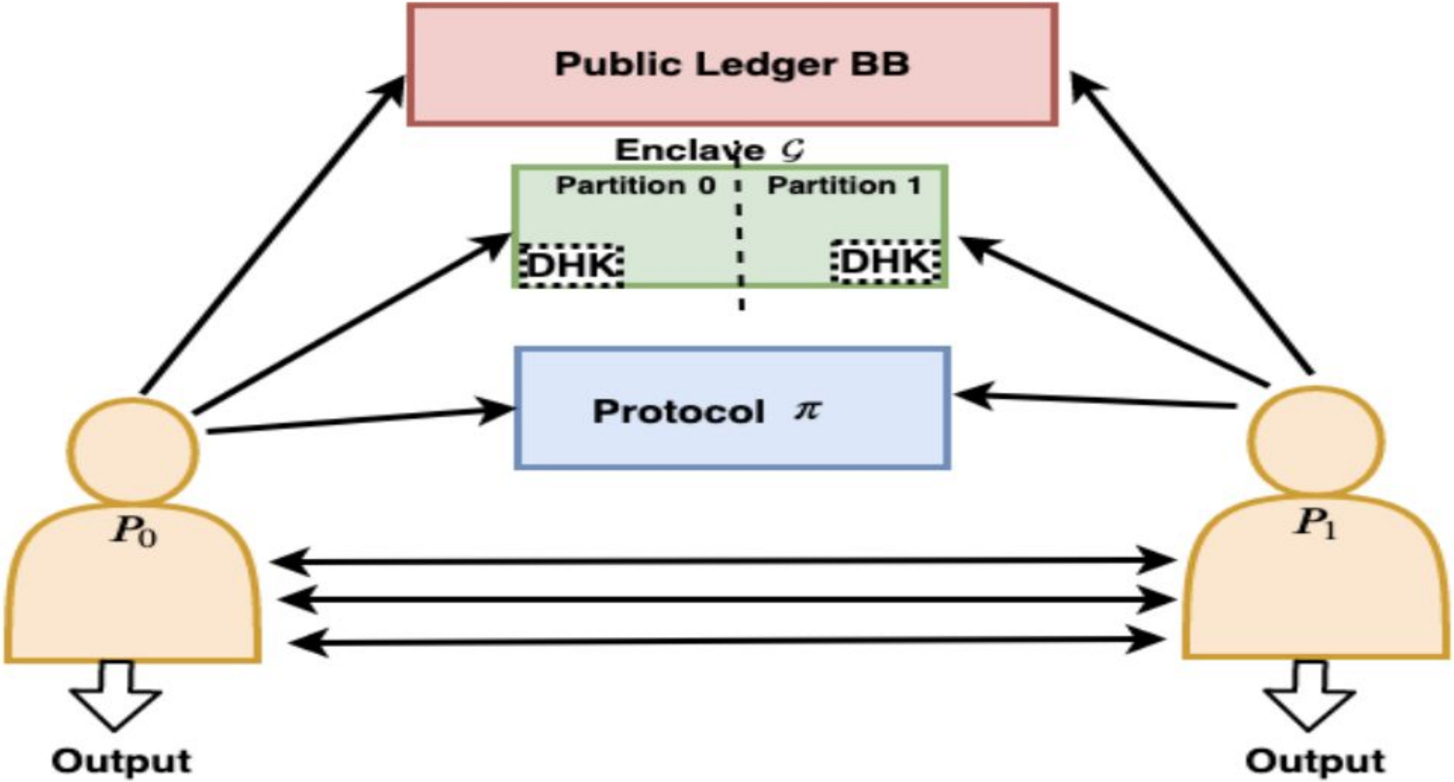
- It provides the private regions of memory -- known as *enclaves* -- for running programs.
- An *enclave* provides *confidentiality* and *integrity* of a program in the presence of adversarial environment.
- It provides attestation of the correct execution of a program using digital signatures.
- Example: Intel Software Guard Extension (SGX)

Component 3: Core MPC having *privacy* and *correctness* security



Here, ct= AE.Enc((k_0, k_1), f(x,y))

Generic Structure of the Protocol



Fair MPC Protocol using BB and Trusted Hardware: CGJ+ Protocol[1]

P_0

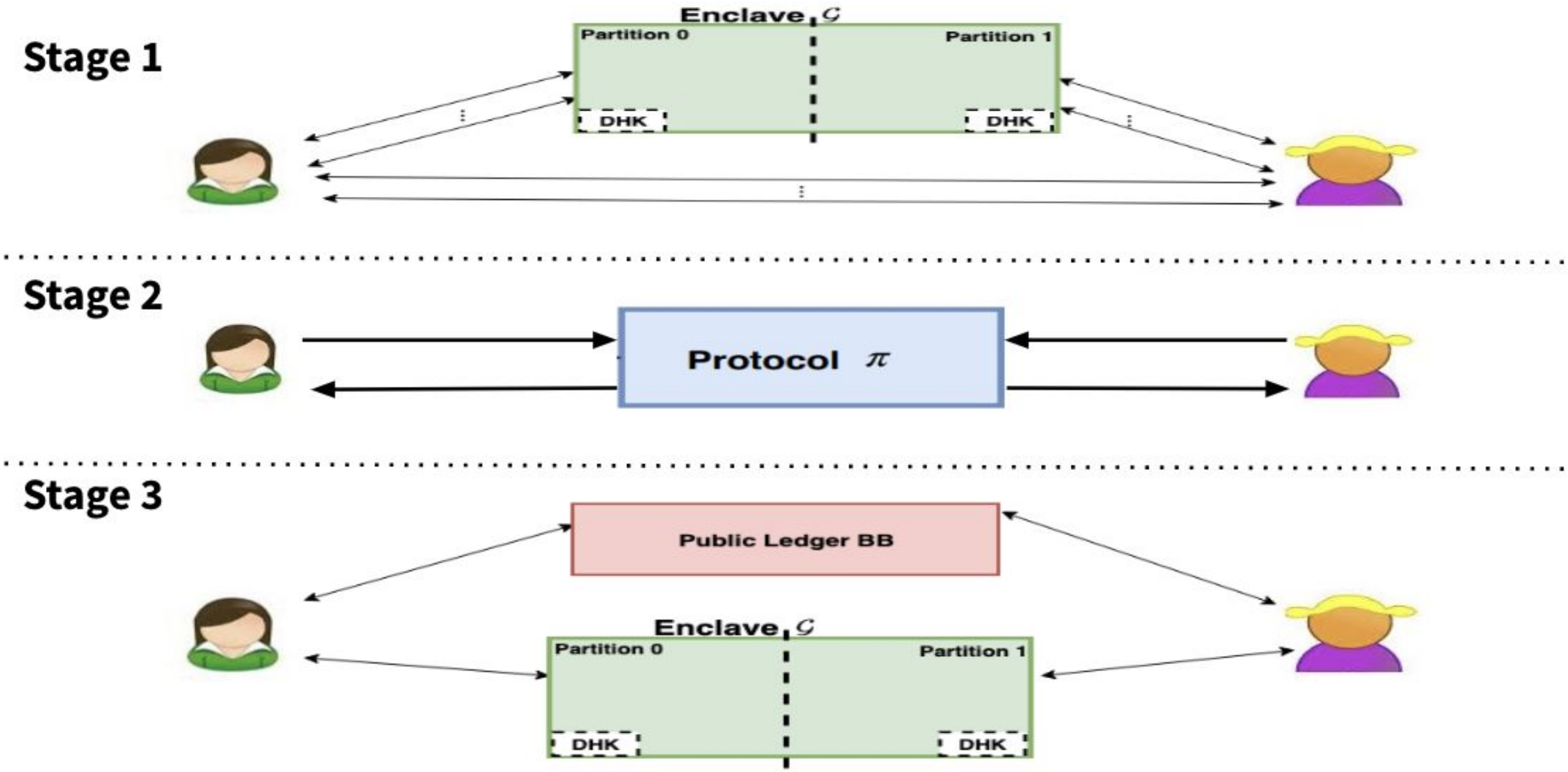
P_1

Secrets: x y

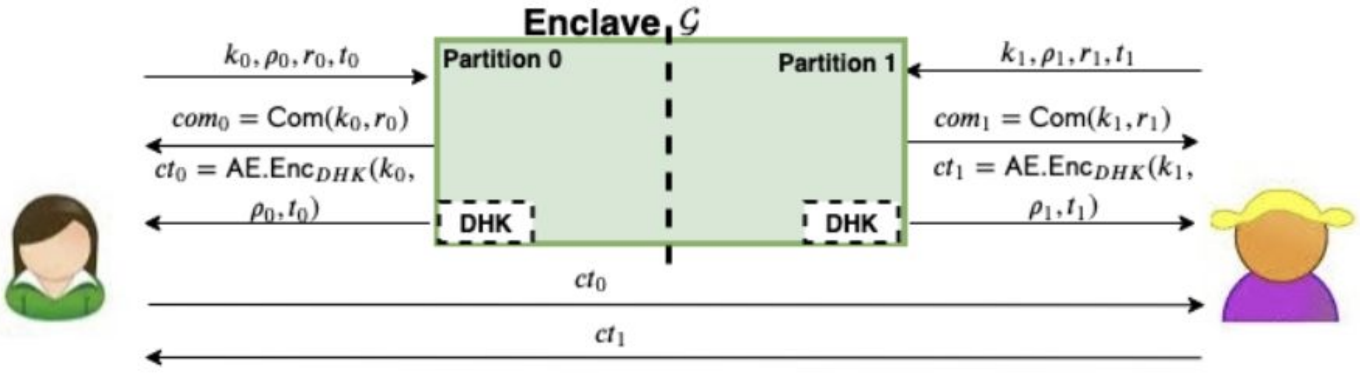
Compute: $f(x,y)$

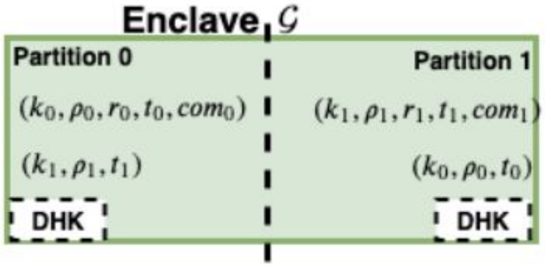
[1]Choudhuri, Arka Rai, et al. "Fairness in an unfair world: Fair multiparty computation from public bulletin boards." *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017.

CGJ+ Protocol

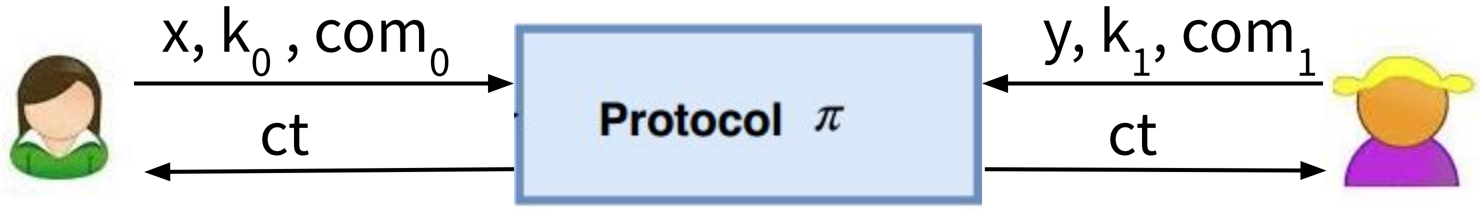


CGJ+ Protocol: Stage 1

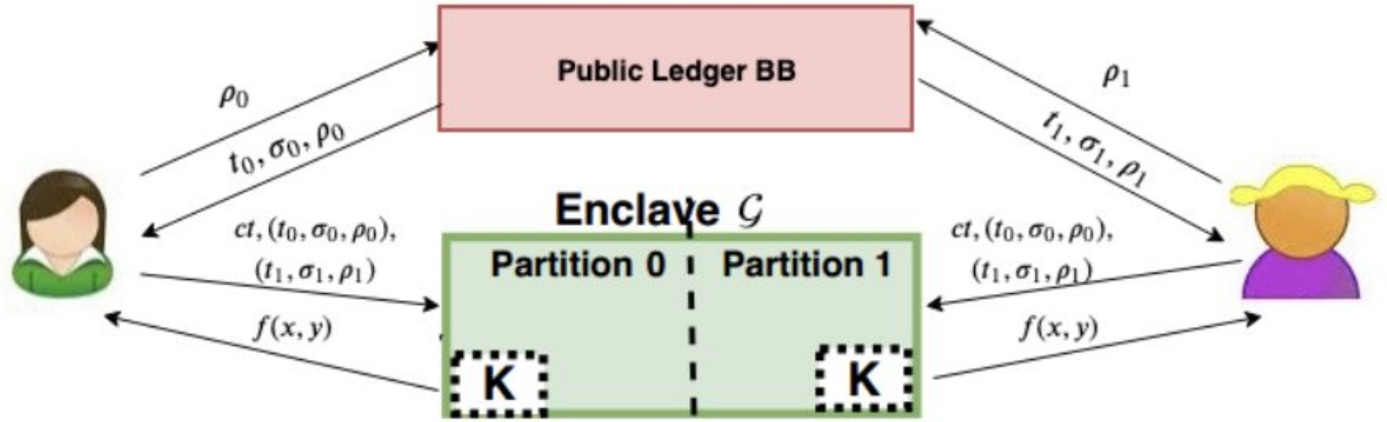




CGJ+ Protocol: Stage 2



CGJ+ Protocol: Stage 3

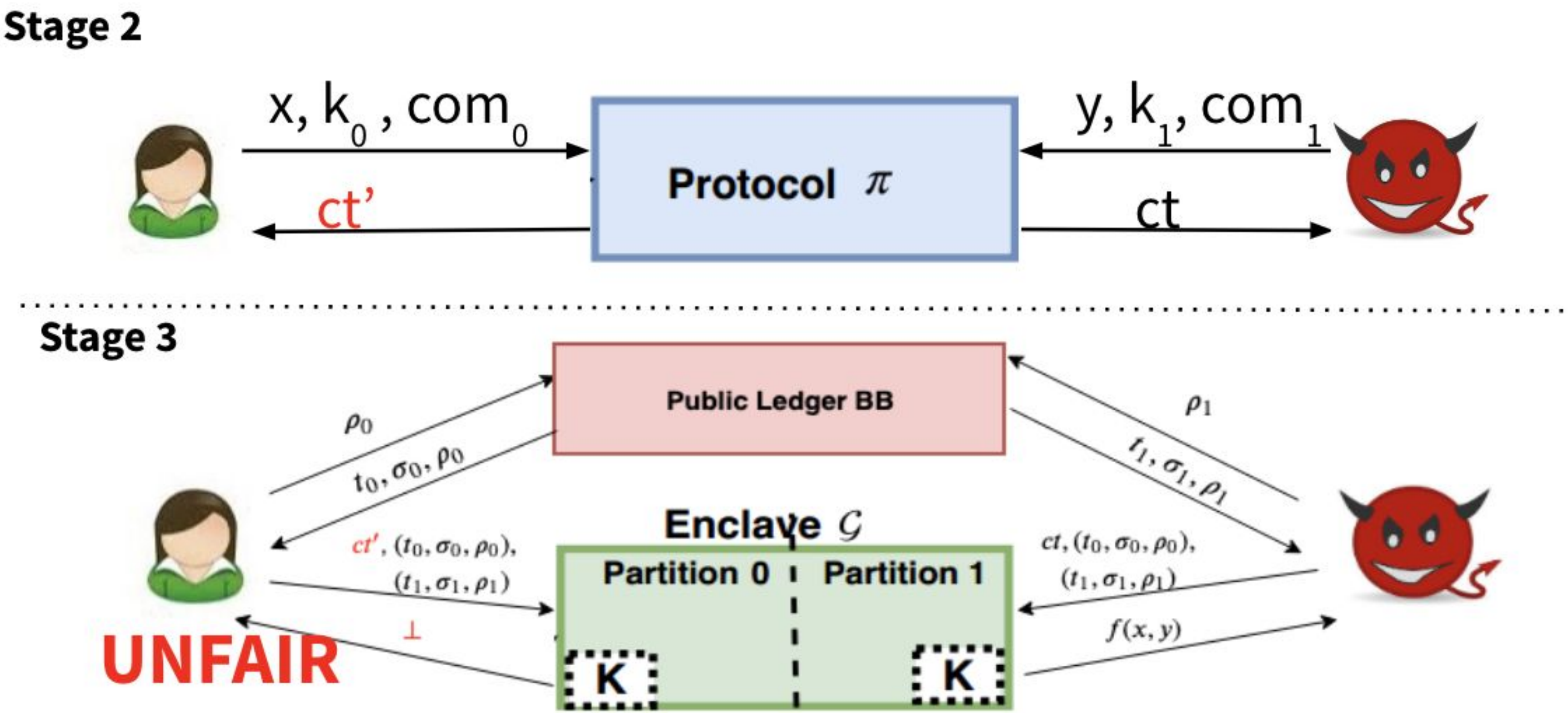


Our Observation

- The security of CGJ+ protocol is proved (in the malicious model with dishonest majority) under the condition that the core MPC component π supports the *privacy* of the individual secrets, and the *correctness* of the output.
- While *privacy* is ensured using a *secret-sharing* scheme, achieving *correctness* of output requires expensive operations such as ZKP and commitment schemes.

Can we break the fairness property of the CGJ+ protocol, if the core MPC component π is allowed to output an incorrect value?

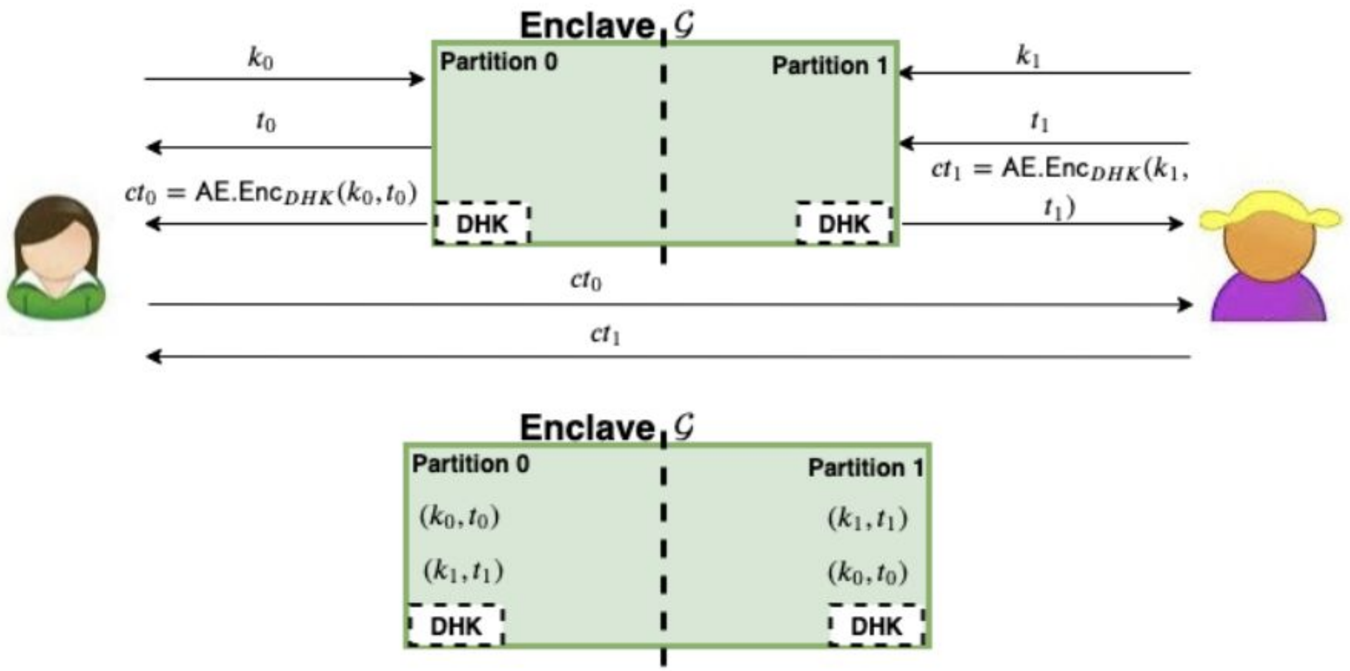
Fairness Attack on CGJ+ Protocol



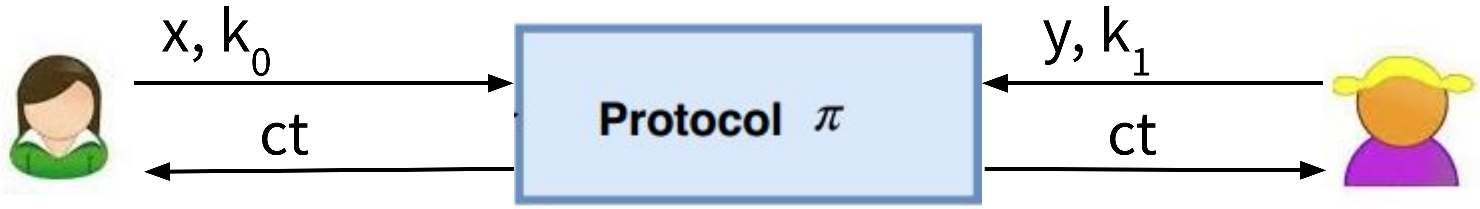
Our Construction

- Designed a new fair protocol Γ, which works even if the internal component π returns an incorrect value.
- We reiterate that the origin of the attack in CGJ+ protocol is the *release tokens* (ρ_0, ρ_1) being generated independently of the ciphertext.
- We remove the r*elease tokens* altogether from the protocol and generate a tag from BB using the ciphertext directly.

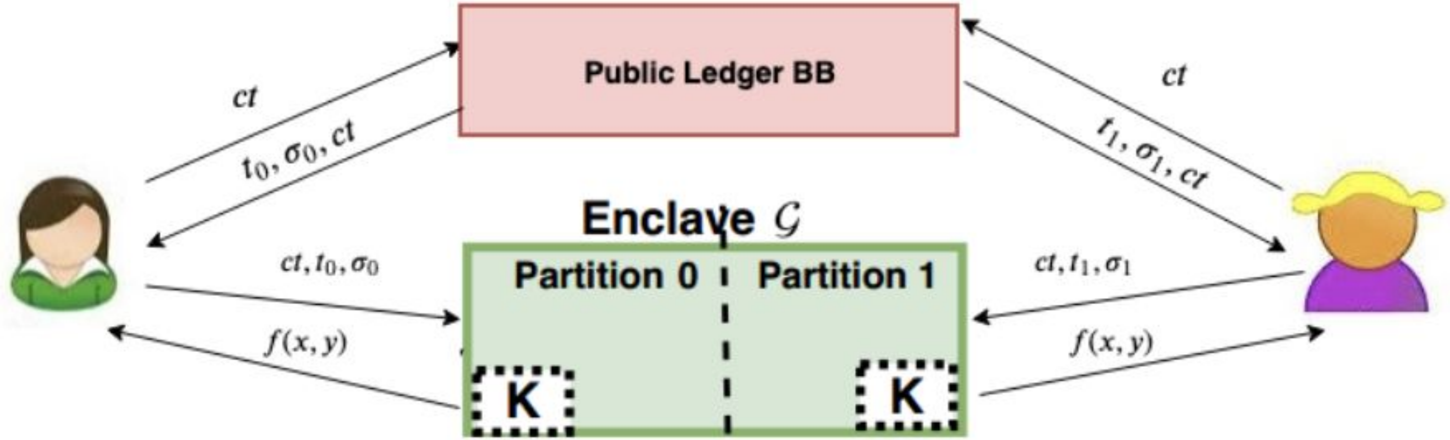
Our Construction: Stage 1



Our Construction: Stage 2



Our Construction: Stage 3



Summary of Our Contribution

- Our first contribution is showing concrete *fairness* attacks on the protocols described in CGJ+, denoted by Π, and KMG[2] (stateless version of CGJ+) protocols, when the underlying protocol π allows incorrect output to be returned.
- Next, we design a new protocol Γ based on public ledger and trusted hardware, and prove that it is *fair*, even if π returns an incorrect value.
- We extended our work to design a stateless version of Γ, namely Υ, and also prove its *fairness*.

[2]Kaptchuk, Gabriel, Matthew Green, and Ian Miers. "Giving State to the Stateless: Augmenting Trustworthy Computation with Ledgers." *NDSS*, 2019.

Results

Protocol	Stateful/ Stateless	Primitives used in π	ZKPoPK amortized compl.	π security		# of var. in $\mathcal{G}$	# of calls in $\mathcal{G}$
				Def. 3	Def. 4		
Π	Stateful	SSS + AE + MAC + ZKPoPK	$O(k+\lambda)$ bits	Fair	Attack	13	Comm.: 1 Enc.: 1 Dec.: 2 OWF: 2
Γ	**Stateful**	SSS + AE	0 bits	**Fair**	**Fair**	**8**	**Comm.: 0** **Enc.: 1** **Dec.: 2** **OWF: 0**
KMG	Stateless	SSS + AE + MAC + ZKPoPK	$O(k+\lambda)$ bits	Fair	Attack	2	Comm.: 2 Encr.: 2 Dec.: 3 OWF: 2 PRF: 2 Hash: 3
Υ	**Stateless**	SSS + AE	0 bits	**Fair**	**Fair**	**2**	**Comm.: 1** **Enc.: 2** **Dec.: 3** **OWF: 0** **PRF: 2** **Hash: 3**

Thank you.