Tight Bounds for Protocols with Hybrid Security

Matthias Fitzi and Dominik Raub

ETH Zürich
Department of Computer Science
CH-8092 Zürich, Switzerland
{fitzi,raubd}(at)inf.ethz.ch

Abstract. We define hybrid multi-party computation (HMPC) and hybrid broadcast (HBC) in a model without broadcast channels but assuming a signature scheme and a respective public-key infrastructure (PKI) among the players. The goal is to achieve unconditional (PKI- and signature-independent) security up to a certain threshold, and security beyond this threshold under stronger assumptions, namely, that forgery of signatures is impossible and/or that the given PKI is consistent. We give a tight characterization of when HMPC and HBC are possible.

1 Introduction

In [LRM08] an optimal trade-off between full information-theoretic (IT) security and computational privacy is proven for multi-party computation (MPC) in a setting where a broadcast channel is available. It appears natural to ask what can be achieved in a setting where the broadcast channel is absent. For the case without additional resources this question has been answered in [FHHW03] and [LRM08]. In this paper we discuss another very natural setting where a digital (pseudo-)signature scheme and a respective public-key infrastructure (PKI) are provided — but no broadcast channels. This setting was treated in [FHW04] with respect to three different levels of security to be achieved simultaneously. In this setting, protocols are defined with respect to three thresholds t_σ, t_p, and T, where $t_\sigma, t_p \leq T$. A protocol is said to achieve *hybrid security* if it is secure under the following condition:

- $t \leq T$ players are corrupted, AND
- if $t > t_\sigma$ players are corrupted then the adversary cannot forge signatures, AND
- if $t > t_p$ players are corrupted then the underlying PKI is consistent.

We call such protocols *hybrid protocols*. In this paper we will provide protocols for hybrid broadcast (HBC) and hybrid multi-party computation (HMPC) — see Sections 4 and 5 for more precise definitions. In the case of broadcast, our protocols will always achieve full-fledged broadcast. In contrast, MPC protocols can only achieve full security (typically characterized by the conditions correctness, privacy, robustness, and fairness) if $T < N/2$ since fairness cannot be achieved beyond [Cle86]. We thus allow the possibility of unfair abort whenever necessary (i.e., exactly under the tight conditions in [LRM08]) — but still abort with agreement, i.e., either no honest player aborts or all honest players do. We show that HBC is possible if and only if

$$T + 2t_\sigma < N \;\wedge\; (t_p > 0 \;\Rightarrow\; 2T + t_p < N) \,, \tag{1}$$

and that, essentially, the same bound is tight for the achievability of HMPC.

2 Model

We assume that the players share a complete synchronous network of pairwise secure channels (for HBC alone, authenticated channels are sufficient). No broadcast channels are available. Instead, the players share a PKI with respect to a given (pseudo-)signature scheme. Still, the PKI might be inconsistent, or the adversary might be able to forge respective signatures.

3 Contribution and Outline

The treatment in [FHW04] is restricted to robust HMPC (implying $T < N/2$) — for which they give a tight bound. In particular, they do not give bounds on the achievability of HBC for $T \geq N/2$. In this paper we give a tight bound for the achievability of HBC for general T, t_σ, and t_p. This result then naturally extends to HMPC.

We first focus on HBC in Section 4 and demonstrate full HMPC in Section 5. In these sections, the protocols are shown to be stand-alone secure. UC security is demonstrated in Section 6. Our bounds are shown to be tight in Section 7.

4 Hybrid Broadcast (HBC)

Definition 1 (BC). *A protocol among a player set P, $|P| = N$, where a player $p_s \in P$ (the sender) inputs a value x_s and every player $p_i \in P$ outputs a value y_i achieves* broadcast (BC) *if the following conditions hold:*

VALIDITY. *If the sender is honest then every honest player p_i outputs $y_i = x_s$.*
CONSISTENCY. *Every honest player p_i outputs the same value $y_i = y$.*
TERMINATION. *All honest players terminate the protocol.* ◇

Definition 2 (HBC). *A protocol among N players with thresholds t_σ, t_p, and T ($t_p, t_\sigma \leq T$) achieves* hybrid broadcast (HBC) *if it achieves broadcast under corruption of t players and the following conditions:*

- *if $t \leq \min(t_p, t_\sigma)$ (unconditionally),*
- *if $(t \leq t_\sigma \wedge t_p < t \leq T)$ and the PKI is consistent,*
- *if $(t \leq t_p \wedge t_\sigma < t \leq T)$ and the adversary cannot forge signatures of honest players.*
- *if $\max(t_\sigma, t_p) < t \leq T$, the PKI is consistent, and the adversary cannot forge signatures of honest players.* ◇

Let us recall the claimed bound for the achievability of HBC:

$$T + 2t_\sigma < n \ \wedge \ (t_p > 0 \ \Rightarrow \ 2T + t_p < n)\ .$$

In order to demonstrate achievability it is sufficient to restrict our treatment to the case where $t_p = 0$ since $t_p > 0$ implies $T < N/2$ — a subcase for which a protocol has already been given in [FHW04]. Another restriction we apply is to assume the underlying PKI to be consistent; and give a generic way how to fix a possibly bad PKI later in Section 4.3.

For our construction of a HBC protocol, a protocol to satisfy the following definition will be constructed as a building block first.

Definition 3 (Broadcast with extended validity (BCEV) [FHHW03]). *A broadcast protocol with sender p_s achieves* broadcast with extended validity *with respect to thresholds t_σ and $T \geq t_\sigma$ if*

t_σ-BROADCAST: *if $t \leq t_\sigma$ players are corrupted the protocol achieves broadcast unconditionally,*

T-VALIDITY: *if $t_\sigma < t \leq T$ players are corrupted and the adversary cannot forge signatures then the protocol achieves the validity condition of broadcast.* ◇

For simplicity, we will restrict our treatment to binary input-message domains in the sequel. Protocols for larger domains can be easily achieved by running binary protocols in parallel.

4.1 An Efficient Protocol for BCEV

Protocol Π_{bcev} is described by the local view of each player p_i (see Figure 1). Let $\sigma_i(x)$ be a signature by player p_i on value x. Let BGP be the efficient, perfectly secure broadcast protocol in [BGP89] tolerating $t < N/3$ corrupted players — it starts with its sender sending his input to every player in the first round.

- In Step 1 of Protocol Π_{bcev}, the sender p_s distributes the pair $(x_s, \sigma_s(x_s))$ where x_s is his input and $\sigma_s(x_s)$ is a signature by p_s on x_s.
- In Step 2, each player p_j redistributes the received pair with an instance of the BGP protocol. Let $(v_i^{j,0}, \sigma_i^{j,0})$ be the initial (first-round) message that p_i receives during the BGP instance with sender p_j, and (v_i^j, σ_i^j) the respective final broadcast result. Player p_i now assembles player sets $S_i^{v,0}$ and S_i^v for each value $v \in \{0,1\}$ — where $j \in S_i^{v,0}$ means that p_j sent v with a valid signature by p_s during the first round of his BGP protocol; and $j \in S_i^v$ means that v together with a valid signature by p_s was received as the final result of p_j's BGP.
- Depending on the cardinalities of the sets $S_i^{v,0}$ and S_i^v, p_i now decides as depicted in Figure 1.

```
1. $p_s$: send $(x_s, \sigma_s(x_s))$.                                   [receive $(x_i, \sigma_i)$]
2. $\forall p_j$: BGP($(x_i, \sigma_i)$).                 [$\forall j$: receive $((v_i^{j,0}, \sigma_i^{j,0}), (v_i^j, \sigma_i^j))$]
   $S_i^{v,0} := \{j | v_i^{j,0} = v \wedge \sigma_i^{j,0}\text{valid}\}$; $S_i^v := \{j | v_i^j = v \wedge \sigma_i^j\text{valid}\}$;
3. if $|S_i^{x_i,0}| \geq N - T \ \wedge \ |S_i^{1-x_i}| = 0$ then $y_i := x_i$                         (I)
   elsif $|S_i^0| > |S_i^1|$ then $y_i := 0$ else $y_i := 1$                                (II)
   fi
```

Fig. 1. Protocol Π_{bcev}.

Lemma 1. *Assuming a consistent PKI, Protocol Π_{bcev} (Figure 1) efficiently achieves BCEV if $T + 2t_\sigma < N$.*

Proof. Efficiency easily follows by inspection of the protocol. Let the number of corrupted players be t. Note that $T + 2t_\sigma < n$ $(t_\sigma \leq T)$ or $t_\sigma = 0$ implies that $t_\sigma < N/3$ and thus that BGP works correctly for $t \leq t_\sigma$.

- **Broadcast ($t \leq t_\sigma$): Validity.** If p_s is honest and $t \leq t_\sigma$ then honest p_i sees $|S_i^{x_s,0}| \geq N - t_\sigma$, $|S_i^{x_s}| \geq N - t_\sigma$, and $|S_i^{1-x_s}| \leq t_\sigma < N - t_\sigma$. Thus p_i decides on $y_i = x_s$ according to either Branches (I) or (II).
- **Broadcast ($t \leq t_\sigma$): Consistency.** If no honest p_i decides according to Branch (I) then all honest p_i decide on the same value since they have the same view of the sets S_i^v. Thus, for the rest, let us assume that some honest p_i decides according to Branch (I). Thus $|S_i^{x_i,0}| \geq N - T$ and $|S_i^{1-x_i}| = 0$. We distinguish two cases.
  1. **honest p_j also decides according to (I).** Since at most t_σ of the players in $S_i^{x_i,0}$ are corrupted, player p_j sees $|S_j^{x_i}| \geq N - T - t_\sigma > 0$. Thus $x_j = 1 - x_i$ is not possible since $S_j^{x_i}$ is not empty. Thus $x_j = x_i$, and consistency follows.
  2. **honest p_j decides according to (II).** Since BGP is reliable ($t_\sigma < N/3$), it follows that $S_j^{1-x_i} = S_i^{1-x_i} = \emptyset$. On the other hand, $S_j^{x_i}$ is not empty: Since there are at least $|S_i^{x_i}| - t \geq N - T - t_\sigma > 0$ honest players $p_h \in S_i^{x_i}$, these players p_h must have sent x_i also during the first round of BGP (see beginning of this section). Thus, p_j sees $|S_j^{x_i}| > 0$, and consistency follows by majority in Branch (II).
- **T-Validity.** It remains to demonstrate validity for $t_\sigma < t \leq T$. Since the adversary cannot forge signatures in this case, an honest player p_i sees $|S_i^{x_s,0}| \geq N - T$ and $S_i^{1-x_s} = \emptyset$ and thus computes $y_i = x_s$. □

4.2 Achieving HBC from BCEV

Protocol $\Pi_{\mathsf{hbc}}^{t_p=0}$ (see Figure 2) is described by the local view of each player p_i. Let p_s be the sender, in_s the sender input, and out_i the broadcast output of player p_i. Let DS-BC denote the efficient broadcast protocol in [DS82] tolerating any number of corrupted players, relying on a PKI and respective signatures.

- In Step 1 of Protocol $\Pi_{\mathsf{hbc}}^{t_p=0}$, the sender p_s distributes his input in_s by an instance of DS-BC.
- In Step 2 the sender distributes his input in_s by an instance of Π_{bcev} (see previous section).
- In Step 3, each player p_i signs his BCEV result and sends the BCEV result together with the signature to every player.[1]
- Now, for each p_i,
  - if some value v was received in Step 3 by at least $N - t_\sigma$ different players p_j together with correct signatures by the p_j then p_i broadcasts the respective signatures with an instance of DS-BC and outputs $out_i = v$;
  - else p_i distributes the empty set with an instance of DS-BC and then computes his output in the following way: If there is a value v such that at least $N - t_\sigma$ valid signatures (originating from different players) were received by some player during this step then $out_i = v$, otherwise p_i accepts the result of the initial DS-BC by the sender as his final output.

```
1. Run Protocol DS-BC on sender input $in_s$.                                  [receive $x_i$]
2. Run Protocol $\Pi_{\mathsf{bcev}}$ on sender input $in_s$.                  [receive $y_i$]
3. Send $y_i$, signed.                                                 [for each $p_j$, receive $z_i^j$]
4. if    some value $v$ was received at least $N - t_\sigma$ times as $z_i^j = v$ for different $j$'s with correct
         signatures then DS-broadcast the respective signatures and $out_i := v$,   [receive $S_i^j$] (I)
   else  DS-broadcast $\emptyset$.                                             [receive $S_i^j$]
         If there is a value $v$ and a set $S_i^j$ consisting of valid signatures on $v$ and $|S_i^j| \geq N - t_\sigma$ then
         $out_i := v$                                                                        (II)
         else $out_i := x_i$                                                                 (III)
         fi
   fi
```

Fig. 2. Protocol $\Pi_{\mathsf{hbc}}^{t_p=0}$.

Lemma 2. *Assuming a consistent PKI, Protocol $\Pi_{\mathsf{hbc}}^{t_p=0}$ efficiently achieves hybrid broadcast if $T + 2t_\sigma < N$.*

Proof. Efficiency easily follows by inspection of the protocol. Let the number of corrupted players be t.

- **$t \leq t_\sigma$.** Protocol Π_{bcev} achieves broadcast, all $N - t_\sigma$ honest players correctly sign their values y_j and resend them in Step 3. Since $N - t_\sigma > N/2$, every honest player p_i uniquely decides on $out_i = y_i = in_s$ in Step 4.

[1] by using some unique message tag in order to separate these signatures from those in DS-BC

- T **corruptions.** It remains to demonstrate broadcast for the case that $t_\sigma < t \leq T$. Note that the adversary cannot forge signatures of honest players in this case.

  Validity. If honest p_i decides on (I) then value out_i was propagated by at least $N - t_\sigma - T > t_\sigma$ honest players in Step 3, and thus $out_i = in_s$ because of T-validity of BCEV. If honest p_i decides on (II) then at least $N - t_\sigma - T > t_\sigma$ honest players must have signed value out_i (with Step-3 tagging), and thus $out_i = in_s$ because of T-validity of BCEV. If honest p_i decides on (III) then $out_i = in_s$ by validity of DS-BC.

  Consistency. If no honest player decides on (I) then all honest players agree since their decisions are based solely on information that was DS-broadcasted, and thus on a common view. Thus, assume that there is an honest player p_j who decides on (I). Note that p_i cannot decide on (III) because $|S_i^j| \geq N - t_\sigma$.

  1. Honest p_i decides on (I): because of p_j's situation p_i sees at least $N - t_\sigma - T > t_\sigma$ signatures on out_i by honest players, and thus $out_i = out_j$ since value $1 - out_i$ cannot have enough support.
  2. Honest p_i decides on (II): for some p_k, $|S_i^k| \geq N - t_\sigma$, and at least $N - t_\sigma - T > t_\sigma$ honest players signed out_i (but no other value with the message tag for Step 3). Thus less than $N - t_\sigma$ players ever signed value $1 - out_i$, and thus $out_j = out_i$. □

4.3 Fixing the PKI

In the previous section we assumed that the given PKI was always consistent. However, since $t_p = 0$, this is not guaranteed in our model for the case that $t = 0$ players are corrupted, i.e., that all players are honest. We now fix this by giving a generic construction on how to transform a possibly bad PKI into a consistent one under the condition that $t_p = 0$. The transformation is based on the protocol FGHHS in [FGH$^+$02] for detectable precomputation of a PKI tolerating any number of corrupted players. This protocol sets up a (possibly inconsistent) PKI with the additional properties that

- either all honest players accept the protocol outcome, or all honest players reject;
- if no player is corrupted then all players accept the protocol outcome;
- if the players accept the protocol outcome then the PKI is consistent.

The transformation works as follows: Let $\mathcal{PKI}$ be the given PKI that might be inconsistent. Before executing the HBC protocol, an instance of FGHHS is executed resulting in an alternative PKI instance $\mathcal{PKI}'$ that might also be inconsistent.

However, if all honest players accept then the new PKI $\mathcal{PKI}$ is consistent and the players can use the given HBC protocol together with the new PKI instead. If the honest players reject then some player must be corrupted ($t > 0$), and the original PKI $\mathcal{PKI}$ is consistent by assumption. In this case, the players run the protocol together with the original PKI.

Lemma 3. *HBC is efficiently achievable if*

$$T + 2t_\sigma < N \ \wedge\ (t_p > 0 \ \Rightarrow\ 2T + t_p < N)\ .$$

Proof. The case $t_p > 0$ follows from [FHW04]. The case $t_p = 0$ follows from Lem. 2 and the discussion of this section. □

We denote the HBC protocol achieving the bound above by Π_{hbc}. Protocol Π_{hbc}, for $t_{\mathsf{PKI}} = 0$, runs FGHHS and $\Pi_{\mathsf{hbc}}^{t_p=0}$, and for $t_{\mathsf{PKI}} > 0$ runs the BC protocol from [FHW04].

Theorem 1. *HBC is (efficiently) achievable if and only if*

$$T + 2t_\sigma < N \ \wedge\ (t_p > 0 \ \Rightarrow\ 2T + t_p < N)\ .$$

Proof. Achievability follows from Lem. 3. Impossibility beyond the given bounds follows from Lem. 8 in Section 7. □

5 Hybrid Multi-Party Computation (HMPC)

We now consider Hybrid Multi-Party Computation (HMPC). As for HBC, the exact guarantees provided by HMPC depend on the number t of corrupted players. As in the treatment of HBC, we define thresholds t_σ for tolerating adversaries that may forge signatures, and t_p for tolerating inconsistent PKIs. Additionally we introduce a threshold t_c for tolerating computationally unbounded adversaries.[2] Moreover, we define robustness limits along the lines of [LRM08], that qualify which robustness properties we guarantee for a given t: For $t \leq \ell_r$ we guarantee fully secure (robust) MPC, for $t \leq \ell_f$ we guarantee fair secure MPC (with privacy, correctness, agreement, and fair abort), and for $t \leq L$ we guarantee abort secure MPC (with privacy, correctness, agreement, and unfair abort). For $t > L$ we make no security guarantees. The threshold T for basic security is not necessary in this context, as its role has been taken by the limit L.

We will, after proving a first general result, derive more specific results with a substantially reduced number of parameters.

Definition 4 (Hybrid Multi-Party Computation (HMPC)). *Let t_σ, t_p, t_c be thresholds, and ℓ_r, ℓ_f, L be limits. Let a PKI and a complete network of secure channels be given. Consider an adversary corrupting t players in the following adversarial setting:*

1. *if $t > t_c$ then the adversary is computationally bounded, otherwise it may be unbounded,*
2. *if $t > t_\sigma$ then the adversary cannot forge signatures, otherwise it may be able to do so,*
3. *if $t > t_p$ then the PKI is consistent, otherwise it may be inconsistent.*

A protocol then achieves hybrid multi-party computation (HMPC) *if it achieves*

1. *fully secure MPC (privacy, correctness, agreement, and robustness) for $t \leq \ell_r$,*
2. *fair secure MPC (privacy, correctness, agreement, and fair abort) for $\ell_r < t \leq \ell_f$,*
3. *abort secure MPC (privacy, correctness, agreement, and unfair abort) for $\ell_f < t \leq L$.* ◇

In [LRM08], for the model *with broadcast channels*, N-player MPC is defined with respect to a limit $\ell_r = \rho$, and implicitly defined limits ℓ_f, L. In keeping with the above definition of these limits, the MPC protocol π^ρ_{SA} of [LRM08] is fully secure for $t \leq \ell_r$, fair secure for $t \leq \ell_f$, and abort secure for $t \leq L$ corrupted parties.[3]

They demonstrate the following bound for the achievability of such MPC:

$$\ell_r \leq \ell_f \leq L \;\wedge\; 2t_c < N \;\wedge\; 2\ell_f < N \;\wedge\; L + \ell_r < N\,. \tag{2}$$

This bound is tight as shown in [IKLP06,Kat07,Cle86,Kil00] and discussed in [LRM08]. Naturally this bound also applies to our weaker model, where only a PKI is provided in place of a BC channel. We will subsequently show that HMPC which is fully secure for $t \leq \max(t_\sigma, t_p)$ is achievable if and only if both the bounds from Eq. (2) and Lem. 3 are satified, i.e.:

Theorem 2 (Bounds for HMPC). *HMPC with thresholds t_σ, t_p, t_c, ℓ_r, ℓ_f, and L that is fully secure for $t \leq \max(t_\sigma, t_p)$ (i.e. where $\ell_r \leq \max(t_\sigma, t_p)$) is achievable if and only if*

$$\begin{aligned} L + 2t_\sigma < N \;\wedge\; (t_p > 0 \;\Rightarrow\; 2L + t_p < N) \qquad \wedge \\ \ell_r \leq \ell_f \leq L \;\wedge\; 2t_c < N \;\wedge\; 2\ell_f < N \;\wedge\; L + \ell_r < N\,. \end{aligned} \tag{3}$$

5.1 Proof of Thm. 2

We first argue the necessity of the bound in Thm. 2. The necessity of the bounds in Eq. (2) inherited from [LRM08] is argued there. Naturally these bounds also apply to our weaker model, where only a PKI is provided in place of a BC channel. It remains to show that the bounds of Lem. 3 are necessary. This does not follow from the fact that they are necessary for HBC. As HMPC only guarantees privacy, correctness, agreement, and unfair abort for $\ell_f < t \leq L$, HMPC

[2] In the preceding sections we could always tolerate computationally unbounded adversaries, as long as they were unable to forge (pseudo-)signatures for $t > t_\sigma$. For the following discussion, we will sometimes need to restrict ourselves to computationally bounded adversaries. Thus, we introduce an additional threshold t_c.

[3] For now, we use the stand-alone secure protocol variant π^ρ_{SA} of the protocol of [LRM08]. The UC setting will be discussed in the next section.

does not imply HBC. Rather it implies non-robust HBC (NRHBC, for short). NRHBC is the same as HBC except except that robustness is not required for $t > \max(t_\sigma, t_p)$. In Sec. 7 we prove that the bounds of Lem. 3 are necessary, not only for HBC but even for NRHBC, thus proving necessity of the bounds in Thm. 2.

We now argue sufficiency, by exhibiting an HMPC protocol that achieves the bounds above. We combine the MPC protocol π^ρ_{SA} of [LRM08] with our HBC protocol Π_{hbc}, thus deriving an MPC protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ for the model with a PKI instead of a BC channel.[3] The MPC protocol π^ρ_{SA} of [LRM08] is fully secure for $t \leq \ell_r$, fair secure for $t \leq \ell_f$, and abort secure for $t \leq L$ corrupted parties. The protocol Π_{hbc} is secure under the bounds of Thm. 1.

In principle, we could choose thresholds for the protocols π^ρ_{SA} and Π_{hbc} independently. The MPC protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ is then secure whenever both the BC protocol Π_{hbc} and the MPC protocol π^ρ_{SA} are secure. In particular, the MPC protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ becomes insecure for $t > \min(L, T)$, as beyond this point either the BC protocol Π_{hbc} (for $t > T$) or the MPC protocol π^ρ_{SA} (for $t > L$) becomes insecure. Hence, lowering the higher of the two parameters to the value of the lower one loses nothing, but gives increased flexibility in choosing the remaining parameters. As such we may without loss set $T = L$. We will for the remainder of this section refer to this parameter as L, as in the definition of HMPC.

Given the fact that the MPC protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ exhibits the security properties of protocol π^ρ_{SA}, whenever it is run with a correct BC channel, we arrive at the following:

Theorem 3 (Security of $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$). *Let a PKI and a complete network of secure channels be given. Let t_σ, t_p, t_c, ℓ_r, ℓ_f, and L be thresholds as in Thm 2*

$$L + 2t_\sigma < N \;\wedge\; (t_p > 0 \Rightarrow 2L + t_p < N) \qquad \wedge \\ \ell_r \leq \ell_f \leq L \;\wedge\; 2t_c < N \;\wedge\; 2\ell_f < N \;\wedge\; L + \ell_r < N\,. \tag{4}$$

Consider a static adversary corrupting t players in the following adversarial setting:

1. *if $t > t_c$ then the adversary is computationally bounded, otherwise it may be unbounded*
2. *if $t_\sigma < t \leq L$ then the adversary cannot forge signatures, otherwise it may be able to do so,*
3. *if $t_p < t \leq L$ then the PKI is consistent, otherwise it may be inconsistent.*

The protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ then achieves hybrid multi-party computation (HMPC). *In other words protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ achieves*

1. *fully secure MPC (privacy, correctness, agreement, and robustness) for $t \leq \ell_r$,*
2. *fair secure MPC (privacy, correctness, agreement, and fair abort) for $\ell_r < t \leq \ell_f$,*
3. *abort secure MPC (privacy, correctness, agreement, and unfair abort) for $\ell_f < t \leq L$.* ◇

Proof. Thm. 3 follows directly from the properties of the MPC protocol π^ρ and of the HBC protocol Π_{hbc} as discussed above. □

Showing that protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ is a secure HMPC protocol achieving the bounds of Eq. (4) completes the proof of Thm. 2. □

5.2 Simplifications

We may simplify the statement of Thm. 3 considerably, by varying just one parameter and deriving the remaining so that they are maximal under the bounds of Eq. (4). The parameters t_c and ℓ_f are independent of all other parameters and we may fix them to their maximal possible value $t_c = \ell_f = \left\lfloor \frac{N-1}{2} \right\rfloor$. We now consider two cases: $t_p = 0$ and $t_p > 0$.

For $t_p > 0$ we have $L < \frac{N}{2}$ by Eq. (4). We can by the bounds in Eq. (4) maximally set $\ell_r = L$. We thus arrive at an MPC protocol with properties as in [FHW04]:

Corollary 1 (Security of $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ for $t_p > 0$). *Let a PKI and a complete network of secure channels be given. Let t_σ, t_p, and L be thresholds such that*

$$L + 2t_\sigma < N \;\wedge\; 2L + t_p < N\,. \tag{5}$$

Consider an unbounded static adversary corrupting t players in the following adversarial setting:

1. *if $t_\sigma < t \leq L$ then the adversary cannot forge signatures, otherwise it may be able to do so,*
2. *if $t_p < t \leq L$ then the PKI is consistent, otherwise it may be inconsistent.*

The protocol $\pi^{\rho}_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ then achieves fully secure MPC for $t \leq L$.

For $t_p = 0$ we obtain new results that go beyond those of [FHW04]. Let us take t_σ as a free parameter. As there are no guarantees for $t_\sigma > L$, we may assume $t_\sigma \leq L$. By Eq. (4), then we have $t_\sigma < \frac{N}{3}$. Maximizing the remaining thresholds under the bounds of Eq. (4) we arrive at the following:

Corollary 2 (Security of $\pi^{\rho}_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ for $t_p = 0$). *Let $t_\sigma < \frac{N}{3}$ and let a PKI (guaranteed to be consistent for $t > 0$ as $t_p = 0$) and a complete and synchronous network of secure channels be given. Consider an unbounded static adversary corrupting t players that, for $t_\sigma < t \leq L$, cannot forge signatures (otherwise it may be able to do so). MPC protocol $\pi^{\rho}_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ is then*

1. *fully secure for $t \leq \min(2t_\sigma, N - 2t_\sigma - 1)$,*
2. *secure with fairness (but without robustness) for $t < \min(\frac{N}{2}, N - 2t_\sigma)$,*
3. *secure with agreement on abort (but without fairness) for $t < N - 2t_\sigma$.*

To see that Cor. 2 provides results beyond those of [FHW04], consider, e.g., the choice $t_\sigma = \frac{N}{5}$, resulting in a protocol that provides security with agreement on abort (but without fairness) for $t < N - 2t_\sigma = \frac{3}{5}N$. As such we provide guarantees exceeding the limit $t < \frac{N}{2}$ of [FHW04].

6 UC Security

We now place our result into the context of universally composable (UC) security.

6.1 Security Definitions and Notations

We follow the Universal Composability (UC) paradigm [PW00,Can01,BPW04][4], which defines a simulation-based security model. The security of a protocol (the real world) is defined with respect to an ideal world, where the computation is performed by a *Trusted Third Party* or *Ideal Functionality* F. Informally, a protocol π achieves security if whatever an adversary can achieve in the real world could also be achieved in the ideal world.

More precisely, let $\mathcal{P} = \{\mathsf{P}_1, \dots, \mathsf{P}_N\}$ be the set of players, and define $[N] := \{1, \dots, N\}$ Then, in the *real world*, there is a given set of resources R (e.g., authenticated or secure channels, BC channels, a PKI) and for each honest player P_i a protocol machine π_i is connected to the resources R. Let $\mathcal{H} \subseteq \mathcal{P}$ denote the set of such honest players. Corrupted players P_i access the resources directly. Let $\mathcal{A} = \mathcal{P} \setminus \mathcal{H}$ denote the set of corrupted players. The *ideal world* consists of the ideal functionality F and an ideal adversary (or simulator) S connected to F.

A protocol π achieves security if, for every possible set of corrupted players $\mathcal{A}$, there is a simulator S such that no environment or distinguisher D can tell the real world and the ideal world apart.[5] For this purpose, the distinguisher directly interacts with either one of the two systems, and in the end outputs a decision bit.

In contrast to [Can01] we use a synchronous communication model with static corruption. As resources R we will generally assume a complete network Net^N of synchronous secure channels.[6]

In this model, a strong composition theorem can be proven [PW00,Can01,BPW04]. In other words, UC security states that wherever a protocol π is used, we can indistinguishably replace this protocol by the corresponding ideal functionality F together with an appropriate simulator. This follows from the free interaction between the distinguisher and the system during the execution, which implicitly models that outputs of the system can be used in arbitrary other protocols, even before the execution ends.[7]

[4] We follow the UC model of [Can01] in spirit, but do not adhere to the notation of [Can01].

[5] In this model, the adversary is thought of as being part of the distinguisher. Canetti [Can01] shows that this is equivalent since the security definition quantifies over all distinguishers.

[6] In [Can01], resources R are modeled as ideal functionalities available in a hybrid model.

[7] This is in contrast to a stand-alone definition of security where the distinguisher is restricted to providing input in the beginning of the computation, and receiving output only at the end.

Definition 5 (Universally Composable (UC) Security). *A protocol π UC securely implements an ideal functionality* F *if* $\forall\mathcal{A},\ \exists\mathsf{S}_{\mathcal{A}},\ \forall\mathsf{D}:\ |Pr[\mathsf{D}(\mathsf{S}_{\mathcal{A}}(\mathsf{F}))=1]-Pr[\mathsf{D}(\pi_{\mathcal{H}}(\mathsf{R}))=1]|\leq\varepsilon(\kappa)$. *Here $\varepsilon(\kappa)$ denotes a negligible function in the security parameter κ,* F *denotes the ideal functionality to be implemented, $\pi_{\mathcal{H}}$ denotes the protocol machines of the honest players in $\mathcal{H}$, and* R *denotes the resources available to the protocol machines. If we admit computationally unbounded distinguishers we obtain information-theoretic (IT) security, if we restrict ourselves to efficient distinguishers and simulators we arrive at computational (CO) security.*

We generally only consider efficient simulators, since otherwise, IT security does not imply CO security. We discuss hybrid-secure protocols that provide different security properties depending on the number of corrupted players and on the computational setting. As such we will use corruption and computational model-aware functionalities that exhibit different behavior depending on the number t of corrupted players and on the computational setting (bounded or unbounded adversaries). We will say that a protocol π UC securely implements an ideal functionality F if π securely implements F in both the CO and the IT setting.

We will, in the following, generally be interested in MPC, i.e., in securely implementing an arbitrary N-player functionality F. We thus model implementing a functionality F with subsets of the security properties privacy, correctness, robustness, fairness, and agreement on abort. We describe the following four specific security notions:

Full Security. Computing functionality F with *privacy, correctness and robustness*, which implies all the security notions mentioned above, is modeled by functionality F itself, since, in the setting which we consider, demanding a secure implementation of functionality F already amounts to demanding full security.

Fair Security. Demanding *privacy, correctness and fairness* (which implies agreement on abort) only for functionality F is captured by the ideal functionality $\mathsf{F}^{\mathsf{fair}}$, which operates as follows: $\mathsf{F}^{\mathsf{fair}}$ internally runs F. Any inputs to F are forwarded, as are any messages F may output to the adversary. If F makes an output y, then $\mathsf{F}^{\mathsf{fair}}$ request an output flag $o\in\{0,1\}$ from the adversary, defaulting to $o=1$ if the adversary makes no suitable input. Finally, for $o=1$ functionality $\mathsf{F}^{\mathsf{fair}}$ makes output y to *all* players, for $o=0$ it halts.

Abort Security. The functionality F^{ab}, specifying *privacy, correctness and agreement on abort* only, works like $\mathsf{F}^{\mathsf{fair}}$ but forwards output y to the adversary before requesting an output flag.[8]

No Security. The functionality $\mathsf{F}^{\mathsf{noSec}}$ models demanding no security whatsoever: Functionality $\mathsf{F}^{\mathsf{noSec}}$ turns control over to the adversary by forwarding all inputs from the honest players to the adversary and letting the adversary fix all outputs to honest players.

As a simulator $\mathsf{S}^{\mathsf{noSec}}$ can use the inputs of honest players to simulate honest protocol machines, this already proves the following (rather trivial) lemma:

Lemma 4. *Any protocol π UC securely implements the ideal model* $\mathsf{F}^{\mathsf{noSec}}$.

6.2 UC Security of HBC

In our synchronous variant of the UC setting, BC can be formalized by means of an ideal BC functionality bc, which behaves as follows: When an arbitrary player P_s gives input x_s, functionality bc outputs (x_s,s) to all players.

Similarly we can formalize HBC using a functionality $\mathsf{bc}^{t_\sigma,t_p,T}$: The behavior of functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ depends on the number of corrupted players t and adversarial setting: Functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ gives up and turns over control to the adversary,

- if $t>T$ or
- if $t>t_\sigma$ and the adversary can forge signatures or
- if $t>t_p$ and the PKI is inconsistent.

Otherwise functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ behaves exactly like the plain BC functionality bc

Lemma 5 (UC security of HBC). *Protocol Π_{hbc} efficiently implements functionality* $\mathsf{bc}^{t_\sigma,t_p,T}$ *whenever*

$$T+2t_\sigma<N\ \wedge\ (t_p>0\ \Rightarrow\ 2T+t_p<N)\,.$$

[8] We could relax the definition further by allowing the adversary to send one output flag for each player, dropping agreement on abort. However, all our protocols will achieve agreement on abort.

Proof of Lem. 5 Efficiency easily follows by inspection of the protocols. We show that the protocol Π_{hbc} indeed implements functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ whenever $T+2t_\sigma < N \;\wedge\; (t_p > 0 \;\Rightarrow\; 2T+t_p < N)$ by providing an appropriate simulator S^{bc}:

The simulator S^{bc} connects to the interfaces of corrupted players to functionality $\mathsf{bc}^{t_\sigma,t_p,T}$. In any situation where functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ turns over control to the adversary simulation is trivial and we have nothing to show. Otherwise, simulator S^{bc} internally emulates the protocol machines Π^i_{hbc} for the honest players P_i and an instance of the PKI. The connections to corrupted players are exposed the adversary.

Let P_i be the honest player with the smallest index i, and P_j be the corrupted player with the smallest index j.

If the internally emulated protocol machine of player P_i outputs (x_s, s) where P_s is corrupted, then S^{bc} inputs x_s to $\mathsf{bc}^{t_\sigma,t_p,T}$ via the interface of P_s.

The simulator S^{bc} identically emulates the same protocol machines Π^i_{hbc} in the ideal model that the honest players run in the real model. This means ideal and real model are perfectly indistinguishable, as long as the outputs of all emulated protocol machines match the outputs of the ideal functionality $\mathsf{bc}^{t_\sigma,t_p,T}$. This amounts to nothing else then demanding consistency and validity as proven above.

6.3 UC Secure HMPC

We now translate Sec. 5 to the UC setting. First, we formalize HMPC by providing an ideal functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$. This functionality evaluates an arbitrary N-player functionality F with the HMPC properties:

Definition 6 (Functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$). *Given an arbitrary N-player functionality F, functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$ behaves as follows:*

1. *If $t > t_c$ and the adversary is computationally unbounded, or*
2. *if $t > t_\sigma$ and the adversary can forge signatures, or*
3. *if $t > t_p$ and the PKI is inconsistent, or*
4. *if $t > L$*

functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$ turns over control to the adversary by running $\mathsf{F}^{\mathsf{noSec}}$. Otherwise functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$ behaves like

1. *functionality F (full security) for $t \le \ell_r$,*
2. *functionality $\mathsf{F}^{\mathsf{fair}}$ (fair security) for $\ell_r < t \le \ell_f$,*
3. *functionality F^{ab} (abort security) for $\ell_f < t \le L$.* ◇

Consider the protocol $\pi^\rho \circ \Pi_{\mathsf{hbc}}$ obtained from the MPC protocol π^ρ of [LRM08] by using out HBC protocol Π_{hbc} for broadcasts. As for the stand-alone setting in Sec. 5 we now show that protocol $\pi^\rho \circ \Pi_{\mathsf{hbc}}$ is a UC secure HMPC protocol in the following sense: Protocol $\pi^\rho \circ \Pi_{\mathsf{hbc}}$ can implement an HMPC for an arbitrary N-player functionality F under the bounds of Eq. (4).

To avoid the impossibility results of [Can01,CF01], we have to move to the CRS-model where a common reference string CRS drawn from a prescribed distribution is made available to all players. So, we will generally assume as resources R a common reference string CRS and a complete network Net^N of synchronous secure channels.A correctly chosen CRS is a prerequisite to the security of the protocols from [LRM08].[9]

Theorem 4 (Security of $\pi^\rho \circ \Pi_{\mathsf{hbc}}$). *Given an arbitrary N-player functionality F, protocol $\pi^\rho \circ \Pi_{\mathsf{hbc}}$ UC securely implements functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$ from a complete network of synchronous secure channels and a CRS for any choice of thresholds respecting*

$$\begin{aligned} L+2t_\sigma < N \;\wedge\; (t_p > 0 \;\Rightarrow\; 2L+t_p < N) \qquad \wedge \\ \ell_r \le \ell_f \le L \;\wedge\; 2t_c < N \;\wedge\; 2\ell_f < N \;\wedge\; L+\ell_r < N\,. \end{aligned} \tag{6}$$

in presence of an active, static adversary.

[9] As noted in [LRM08], it is possible to minimize the reliance on the CRS such that our protocols tolerate an adversarially chosen CRS for few corrupted players by applying techniques from [GK08,GO07] and a $(t, 2t-1)$-combiner for commitments (e.g. [Her05]). However, this construction is beyond the scope of this paper.

Proof. The proof of Thm. 4 is almost trivial. By Lem. 5 the BC protocol Π_{hbc} implements functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ under the bounds of Lem. 3. According to the UC Theorem it is hence sufficient to prove that protocol $\pi^\rho \circ \mathsf{bc}^{t_\sigma,t_p,L}$ implements functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$. Now, by the choice of bounds in the definition of functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$, we find: In any setting where functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$ does not turn over control to the adversary, the functionality $\mathsf{bc}^{t_\sigma,t_p,T}$ behaves exactly like a plain bc. But on a plain bc, protocol π^ρ precisely the guarantees made by functionality $\mathsf{F}^{\mathsf{hyb}}_{t_\sigma,t_p,t_c,\ell_r,\ell_f,L}$ (also see [LRM08]). □

Results along the lines of Cor. 1 and Cor. 2 are easily translated to the UC setting. We refrain from restating them here.

7 Impossibility

We demonstrate impossibility of HMPC beyond the bounds of Lem. 2 by showing that the special case of an HMPC, namely (unfair) non-robust HBC (NRHBC, for short), is not achievable. NRHBC is the same as HBC except that robustness is not required for $t > \max(t_\sigma, t_p)$.

Definition 7 (NRBC). *A protocol among a player set P, $|P| = N$, where a player $p_s \in P$ (the sender) inputs a value $x_s \neq \perp$ and every player $p_i \in P$ outputs a value y_i achieves* non-robust broadcast (NRBC) *if the following conditions hold:*

VALIDITY. *If the sender is honest then either every honest player p_i outputs $y_i = x_s$ or every honest player p_i outputs $y_i = \perp$.*

CONSISTENCY. *Every honest player p_i outputs the same value (where $\perp$ is an allowed output).*

TERMINATION. *All honest players terminate the protocol.* ◇

Definition 8 (NRHBC). *A protocol among a player set P, $|P| = N$, where a player $p_s \in P$ (the sender) inputs a value $x_s \neq \perp$ and every player $p_i \in P$ outputs a value y_i achieves* non-robust hybrid broadcast (NRHBC) *if the following conditions hold:*

- *if $t \leq \max(t_p, t_\sigma)$ then the protocol achieves broadcast,*
- *if $\max(t_p, t_\sigma) < t \leq T$ then the protocol achieves non-robust broadcast.* ◇

Impossibility of NRHBC beyond the bounds of Lem. 2 can be shown along the lines of [FLM86] and [FHW04] by demonstrating the impossibility of the following two special cases:

1. First, we show that NRHBC is impossible if $t_p = 0$, $t_\sigma > 0$, and $T + 2t_\sigma \geq N$.
2. Second, we show that NRHBC is impossible if $t_\sigma = 0$, $t_p > 0$, and $2T + t_p \geq N$.

7.1 Impossibility of $\mathbf{T + 2t_\sigma \geq N}$ when $\mathbf{t_\sigma > 0}$.

For the sake of contradiction, assume a protocol that achieves NRHBC under these conditions among a player set P, $|P| = N \geq 3$. We can partition the players in to three sets P_0, P_1, and P_2, with cardinalities $|P_0| \leq T$ and $|P_1|, |P_2| \leq t_\sigma$ where the sender p_s is in P_0. Let p'_i be a copy of player $p_i \in P_0$ and $P'_0 = \{p'_i | p_i \in P_0\}$ where player p'_i holds the same protocol information as p_i. We show that the assumed protocol leads to a contradiction when we connect the players in $P' = P_0 \cup P_1 \cup P_2 \cup P'_0$ in a certain way and let the protocol run.

The players are connected in the following way — see Figure 3. Exactly all pairs in $(P_0 \cup P_1) \times (P_0 \cup P_1)$, $(P_1 \cup P_2) \times (P_1 \cup P_2)$, and $(P_2 \cup P'_0) \times (P_2 \cup P'_0)$ are connected by pairwise channels meaning that a message that normally would be sent from $p_i \in P_2$ to $p_j \in P_0$ is sent from p_i to $p'_j \in P'_0$ instead, and that p'_j communicates with the players in $P_2 \cup P'_0$ as it would with the players in $P_2 \cup P_0$ under normal conditions. Note that no further connections exist.

We first show that for input $x_s = 0$ of the original sender p_s and input $x'_s = 1$ of the sender's copy p'_s the joint view among the different sets $P_0 \cup P_1$, $P_1 \cup P_2$, and $P_2 \cup P'_0$, are indistinguishable from their joint view in a protocol under normal conditions where the adversary corrupts the remaining players.

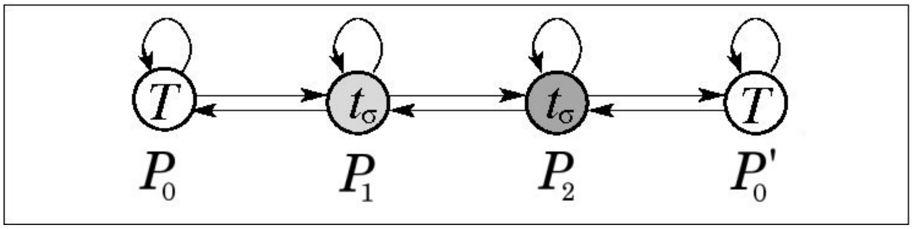

Fig. 3. Simulated system by the adversary for the case $t_\sigma > 0$

Joint view of $P_0 \cup P_1$ *with* $x_s = 0$. By corrupting all players in P_2 in the original system the adversary simulates all players in $P_2 \cup P_0'$ of the new system. Since $|P_2| \leq t_\sigma$, the adversary can forge all signatures by players in P_0' required for the simulation. Thus the joint view of the players in $P_0 \cup P_1$ in the original system is exactly the same as their view in the new system.

Joint view of $P_2 \cup P_0'$ *with* $x_s' = 1$. By symmetry, this case follows from the above paragraph.

Joint view of $P_1 \cup P_2$. By corrupting all players in P_0 in the original system the adversary can simulate all players in $P_0 \cup P_0'$ of the new system. Note that, by corrupting the players in P_0, the adversary gains access to all corresponding secret keys and thus is not required to forge any signatures for the simulation. Thus the joint view of the players in $P_1 \cup P_2$ in the original system is exactly the same as their view in the new system.

Contradiction. The assumption that the given protocol achieves NRHBC implies that the players $p_i \in P_0 \cup P_1$ must agree on $y_i = x_s = 0$ since the adversary corrupting the at most t_σ players in P_2 implies that $t \leq \max(t_\sigma, t_p)$. By the same argument, the players $p_j \in P_2 \cup P_0'$ must agree on $y_j = x_s' = 1$. However, this implies that the players in P_1 and the players in P_2 disagree on their outputs in contradiction to the definition of NRHBC. This implies that NRHBC is not achievable under these conditions.

Lemma 6. *If* $t_\sigma > 0$ *then NRHBC is not achievable if* $T + 2t_\sigma \geq N$.

Proof. The proof follows from the above discussion. □

7.2 Impossibility of $\mathbf{2T + t_p \geq N}$ when $\mathbf{t_p > 0}$.

We proceed in the same way as in the previous section. We can partition the players in to three sets P_0, P_1, and P_2, with cardinalities $|P_0| \leq t_p$ and $|P_1|, |P_2| \leq T$ where the sender p_s is in P_0. Let p_i', P_0', and P' be defined as in the previous section.

The players are connected in the following way — see Figure 4. Exactly all pairs in $(P_0 \cup P_1) \times (P_0 \cup P_1)$, $(P_1 \cup P_2) \times (P_1 \cup P_2)$, and $(P_2 \cup P_0') \times (P_2 \cup P_0')$ are connected as in the previous section. Additionally, for all players $p_i' \in P_0'$ the old secret-key/public-key pair is erased and replaced by a random valid pair $(\mathrm{SK}_i', \mathrm{PK}_i')$; and for all players $p_j \in P_2 \cup P_0'$ p_j's copy of PK_i is replaced by PK_i'.

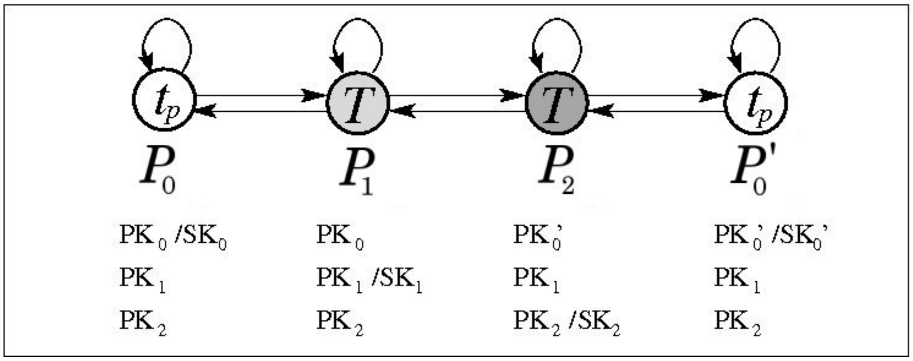

Fig. 4. Simulated system by the adversary for the case $t_p > 0$

We again show that for input $x_s = 0$ of the original sender p_s and input $x'_s = 1$ of the sender's copy p'_s the joint view among the different sets $P_0 \cup P_1$, $P_1 \cup P_2$, and $P_2 \cup P'_0$, are indistinguishable from their joint view in a protocol under normal conditions where the adversary corrupts the remaining players.

Joint view of $P_0 \cup P_1$ with $x_s = 0$. By corrupting all players in P_2 in the original system the adversary simulates all players in $P_2 \cup P'_0$ of the new system. For all $p_i \in P'_0$ it generates a random valid secret-key/public-key pair $(\mathrm{SK}'_i, \mathrm{PK}'_i)$ and overwrites p_i's own secret key, and, for all $p_j \in P_2 \cup P'_0$, overwrites p_j's copy of p_i's public key. The PKI among the players in $P_0 \cup P_1$ is still fully consistent and thus the joint view of the players in $P_0 \cup P_1$ in the original system is exactly the same as their view in the new system.

Joint view of $P_2 \cup P'_0$ with $x'_s = 1$. By symmetry, this case follows from the above case.

Joint view of P_1 and P_2. Since $|P_0| \leq t_p$ the adversary can have previously made the PKI inconsistent by generating and respectively distributing the key pairs $(\mathrm{SK}'_i, \mathrm{PK}'_i)$ for all $p_i \in P'_0$ (according to Figure 4). By corrupting all players in P_0 in the original system the adversary can now simulate all players in $P_0 \cup P'_0$ of the new system. Thus the joint view of the players in $P_1 \cup P_2$ in the original system is exactly the same as their view in the new system.

Contradiction. Assuming the protocol to achieve NRHBC now implies that the players $p_i \in P_0 \cup P_1$ must agree on either $y_i = x_s = 0$ or $y_i = \perp$ since at most T players are corrupted. By the same argumentation, the players $p_j \in P_2 \cup P'_0$ must agree on either $y_j = x'_s = 1$ or $y_j = \perp$. However, this implies that the players in P_1 and the players in P_2 either disagree on their outputs or jointly output $\perp$ — which is not allowed under the corruption of $t \leq t_p$ players. This concludes that NRHBC is not achievable under these conditions.

Lemma 7. *If $t_p > 0$ then NRHBC is not achievable if $2T + t_p \geq N$.*

Proof. The proof follows from the above discussion. □

Lemma 8. *NRHBC, HBC, and HMPC, are not not achievable if $t_\sigma > 0$ and $T + 2t_\sigma \geq N$, or if $t_p > 0$ and $2T + t_p \geq N$.*

Proof. The lemma follows from Lem. 6, Lem. 7, the fact that HBC implies NRHBC, and the fact that NRHBC is an instance of HMPC. □

8 Conclusions

We describe a hybrid broadcast (HBC) protocol $\mathit{\Pi}_{\mathsf{hbc}}^{t_p=0}$ and, building on it, a hybrid MPC (HMPC) protocol $\pi^\rho \circ \mathit{\Pi}_{\mathsf{hbc}}^{t_p=0}$ for a setting where a PKI and a complete network of secure channels, but no broadcast channels are provided. We thereby extend the work of [LRM08] to the setting where a PKI is available instead of a BC channel, and we extend the work of [FHW04] to the setting where robustness is not always required.

Our protocols achieve different levels of security, depending on the number t of corrupted players:

- For $t \leq t_\sigma$ we tolerate adversaries that may forge signatures,
- for $t \leq t_p$ we tolerate inconsistent PKIs,
- for $t \leq T$ we achieve broadcast.

Furthemore we have a number of limits that pertain to HMPC only:

- For $t \leq t_c$ we tolerate computationally unbounded adversaries,
- for $t \leq \ell_r$ we guarantee fully secure MPC (with privacy, correctness, agreement, and robustness),
- for $t \leq \ell_f$ we guarantee fair secure MPC (with privacy, correctness, agreement, and fair abort),
- for $t \leq L$ we guarantee abort secure MPC (with privacy, correctness, agreement, and unfair abort).

We demonstrate that our HBC and HMPC protocols are optimal by showing that HBC is achievable if and only if

$$T + 2t_\sigma < n \;\wedge\; (t_p > 0 \;\Rightarrow\; 2T + t_p < n)\;.$$

and that HMPC (with the minimal robustness requirement $\ell_r \geq \max(t_\sigma, t_p)$) is achievable if and only if

$$\begin{aligned} L + 2t_\sigma < N \;\wedge\; (t_p > 0 \;\Rightarrow\; 2L + t_p < N) \qquad \wedge \\ \ell_r \leq \ell_f \leq L \;\wedge\; 2t_c < N \;\wedge\; 2\ell_f < N \;\wedge\; L + \ell_r < N\;. \end{aligned}$$

References

[BGP89] Piotr Berman, Juan A. Garay, and Kenneth J. Perry. Towards optimal distributed consensus (extended abstract). In *Proceedings of IEEE Symposium on the Foundations of Computer Science (FOCS) '89*, pages 410–415, 1989.

[BPW04] Michael Backes, Birgit Pfitzmann, and Michael Waidner. A general composition theorem for secure reactive systems. In *TCC'04*, volume 2951 of *LNCS*, pages 336–354. Springer, 2004.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO'01*, pages 19–40. Springer, 2001.

[Cle86] R Cleve. Limits on the security of coin flips when half the processors are faulty. In *STOC'86*, pages 364–369. ACM, 1986.

[DS82] Danny Dolev and H. Raymond Strong. Polynomial algorithms for multiple processor agreement. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC) '82*, pages 401–407. ACM, 1982.

[FGH$^+$02] Matthias Fitzi, Daniel Gottesman, Martin Hirt, Thomas Holenstein, and Adam Smith. Detectable Byzantine agreement secure against faulty majorities. In *21st ACM Symposium on Principles of Distributed Computing (PODC)*, pages 118–126, 2002.

[FHHW03] Matthias Fitzi, Martin Hirt, Thomas Holenstein, and Jürg Wullschleger. Two-threshold broadcast and detectable multi-party computation. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 265 of *Lecture Notes in Computer Science*, pages 51–67. Springer-Verlag, May 2003.

[FHW04] Matthias Fitzi, Thomas Holenstein, and Jürg Wullschleger. Multi-party computation with hybrid security. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 419–438. Springer-Verlag, May 2004.

[FLM86] Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. In Barbara B. Simons and Alfred Z. Spector, editors, *Fault-Tolerant Distributed Computing*, volume 448 of *Lecture Notes in Computer Science*, pages 147–170. Springer, 1986.

[GK08] Vipul Goyal and Jonathan Katz. Universally composable multi-party computation with an unreliable common reference string. In *TCC'08*, volume 4948 of *LNCS*, pages 142–154. Springer, 2008.

[GO07] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. In *CRYPTO'07*, volume 4622 of *LNCS*, pages 323–341. Springer, 2007.

[Her05] Amir Herzberg. On tolerant cryptographic constructions. In *CT-RSA'05*, volume 3376 of *LNCS*, pages 172–190. Springer, 2005.

[IKLP06] Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. On combining privacy with guaranteed output delivery in secure multiparty computation. In *CRYPTO'06*, volume 4117/2006, pages 483–500. Springer, 2006.

[Kat07] Jonathan Katz. On achieving the “best of both worlds” in secure multiparty computation. In *STOC'07*, pages 11–20. ACM, 2007.

[Kil00] Joe Kilian. More general completeness theorems for secure two-party computation. In *STOC'00*, pages 316–324. ACM, 2000.

[LRM08] Christoph Lucas, Dominik Raub, and Ueli Maurer. Optimally hybrid-secure mpc. Available at `eprint.iacr.org/2009/009`, 2008.

[PW00] Birgit Pfitzmann and Michael Waidner. Composition and integrity preservation of secure reactive systems. In *ACM CCS'00*, pages 245–254, 2000.