Tight Bounds for Protocols with Hybrid Security

Matthias Fitzi[1] and Dominik Raub[2]

[1] ETH Zürich
Department of Computer Science
CH-8092 Zürich, Switzerland
fitzi(at)inf.ethz.ch

[2] Aarhus University
Department of Computer Science
DK-8000 Aarhus C, Denmark
raub(at)cs.au.dk

Abstract. We consider broadcast and multi-party computation (MPC) in the setting where a digital signature scheme and a respective public-key infrastructure (PKI) are given among the players. However, neither the signature scheme nor the PKI are fully trusted. 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 the forgery of signatures is impossible and/or that the given PKI is not under adversarial control. We give protocols for broadcast and MPC that achieve an optimal trade-off between these different levels of security.

1 Introduction

1.1 Multi-Party Computation

In [Yao82], Yao introduced the concept of secure multi-party computation (MPC): Given an arbitrary but fixed function f and a set of N mutually distrusting players, an MPC protocol allows these players to compute the function f on their inputs securely, even if some of the players are corrupted by an adversary. This first notion (often called secure function evaluation) has meanwhile been extended to reactive and randomized functionalities.

Security requirements for MPC in the literature (e.g. [Gol01]) include privacy, correctness, robustness, fairness, and agreement on abort. *Privacy* is achieved if the adversary cannot learn more about the honest players' inputs than what is implied by the inputs and outputs of the corrupted players. *Correctness* means that the protocol output is correct for the given inputs according to the specification, or that there is no output. In this paper, our notion of security generally encompasses these two basic requirements, privacy and correctness. Possible additional requirements are notions of output guarantees, which we discuss in order of decreasing strength: A protocol achieves *robustness* if the adversary cannot prevent the honest players from obtaining output, i.e., if it is guaranteed that no honest player aborts the protocol. *Fairness* is achieved if the adversary cannot get any information about the honest players' inputs in case that any honest player aborts. *Agreement on abort* means that either all honest players abort or none of them does. Security (privacy and correctness) with robustness is often referred to as *full security*. Accordingly, we use the term *fair security* for privacy, correctness, agreement on abort, and fairness (without robustness), and the term *abort security* for privacy, correctness, and agreement on abort (without robustness or fairness).

1.2 Full Security

A first general solution to the MPC problem was given by Goldreich et al. [GMW87] based on computational intractability assumptions and the availability of a broadcast (BC) channel. They achieve full security against $t < \frac{N}{2}$ actively corrupted players. Ben-Or et al. [BGW88] and Chaum et al. [CCD88] presented protocols which are information-theoretically (IT) secure and require no BC channel. They achieve full security against $t < \frac{N}{3}$ actively corrupted players. When additionally assuming BC channels IT full security can be achieved for up to $t < \frac{N}{2}$ actively corrupted players as was shown by Beaver [Bea89] and Rabin et al. [RB89].

1.3 Hybrid Security

For MPC protocols, reliance on computational intractability assumptions (e.g. difficulty of factoring) or trusted setup (e.g. broadcast or correct PKI) is undesirable. Such assumptions are generally unproven or even unprovable, and their invalidation generally leads to a complete loss of security for protocols based on the assumption, even if only a single player is corrupted. On the other hand, IT secure MPC can only tolerate a relatively small fraction of actively corrupted players. This leads to the natural question whether it is possible to construct protocols that are fully IT secure against a small portion of corrupted players but that, at the same time, still provide some weaker security guarantees for the case that some larger portion of players is corrupted. That is, are there protocols that allow for a graceful degradation of security as the number of corrupted players rises?

For the standard models with a given broadcast channel or without broadcast nor a public-key infrastructure (PKI) a full characterization of such degradation was given in [FHHW03,LRM10].

In [FHW04] another natural model was treated where no broadcast channels are given but a digital (pseudo-)signature scheme and a respective public-key infrastructure (PKI) — but where the PKI might be inconsistent or the adversary might be able to forge respective signatures. In this model, 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 refer to such protocols as *hybrid protocols*. Note that assuming a PKI is a much more realistic assumption than assuming broadcast channels since secure physical broadcast channels do not exist. Furthermore, allowing for a possibly inconsistent PKI or possible forgeries by the adversary makes the model even more realistic.

Gupta et al. [GGBS10] and Gordon et al. [GKKY09] consider a related setting, where the private keys of t_c parties may be compromised while t_a parties may additionally be actively corrupted. In [GKKY09] it is claimed that the model in [FHW04] constitutes the special cases $t_c = 0$ and $t_c = N$ of their model. Note that this is not correct since the model in [FHW04] allows that the adversary *either* corrupts a large number of players but cannot forge signatures *or* corrupts a small number of players while being able to forge signatures; whereas both alternatives are tolerated by the same protocol.

1.4 Contributions

The treatment in [FHW04] was restricted to *fully secure* hybrid MPC (its robustness implying $T < N/2$ by an argument in [Cle86]) — for which they give a tight bound. In particular, they did not give bounds on the achievability of hybrid broadcast for $T \geq N/2$. In this paper we give a tight bound for the achievability of hybrid broadcast for general T, t_σ, and t_p. This result then naturally extends to hybrid MPC for general thresholds. In the case of broadcast, our protocols will always achieve full-fledged broadcast. In the case of MPC, we have to allow the possibility of unfair abort whenever necessary (i.e., exactly under the tight conditions in the model with broadcast [LRM10]) — but still abort with agreement, i.e., either no honest player aborts or all honest players do.

The considered adversary is assumed to be static. Note that adaptive security is provably unachievable for $T > N/2$ [HZ10]. We show that hybrid broadcast is possible if and only if

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

Furthermore, we show that, essentially, Bound (1) is also tight for the achievability of hybrid MPC.

1.5 Model

As is common for MPC we consider a set $\mathcal{P}$ of $|\mathcal{P}| = N$ players connected by a complete network of secure[3] channels. No broadcast channels are available. Instead, the players share a public-key infrastructure (PKI) with respect to a given (pseudo-)signature scheme. When discussing universally composable (UC) MPC protocols we also require a common reference string (CRS) to avoid the impossibility results of [Can01,CF01]. The adversary may actively corrupt an arbitrary subset of t players, taking full control of them.

Furthermore, the public-key infrastructure (PKI) may or may not be controlled by the adversary (and inconsistent, in particular). If not under adversarial control the PKI provides, with respect to each player $\mathsf{P}_i \in \mathcal{P}$, a random signing-key/verification-key(s) pair for a secure[4] (pseudo)-signature scheme such that

[3] As an exception, authenticated channels are sufficient for the special case of broadcast when based on standard (non-IT) signatures.

[4] We require the signature scheme to be existentially unforgeable under adaptive chosen-message attacks (UF-CMA), e.g. [CS99]. Alternatively an IT secure pseudo-signature scheme, e.g. [PW96], can be used.

P_i has exclusive access to his own signing key and each player P_j has access to a verification key for the verification of signatures by P_i. If under adversarial control the adversary receives all inputs to and arbitrarily fixes all outputs by the PKI, i.e., defines all involved keys in any possible way. Additionally, the adversary may be able to forge signatures. We model such adversaries by giving them access to all players' signing keys. More precise UC formalizations can be found in App. A.

Note that the above modelling implies that controlling the PKI is strictly more powerful than the ability to forge signatures. This implies that we can generally assume that $t_p \leq t_\sigma$.

2 Hybrid Broadcast (HBC)

In this section we discuss constructions for hybrid-secure broadcast (HBC). We begin by defining broadcast:

Definition 1 (BC). *A protocol among a player set $\mathcal{P}$ of size $|\mathcal{P}| = N$ where a player $\mathsf{P}_s \in \mathcal{P}$ (the sender) inputs a value x_s and every player $\mathsf{P}_i \in \mathcal{P}$ outputs a value y_i achieves* broadcast (BC) *if the following conditions hold:*

VALIDITY. *If the sender P_s 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.* ◇

A BC protocol is hybrid-secure if it tolerates $t \leq T$ corrupted players, tolerates an adversarially controlled PKI in presence of $t \leq t_p$ corrupted players, and tolerates an adversary capable of forging signatures in presence of $t \leq t_\sigma$ corrupted players:

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

- *if $t \leq t_p$ (unconditionally),*
- *if $t_p < t \leq t_\sigma$ and the PKI is correct (trusted),*
- *if $t_\sigma < t \leq T$, the PKI is correct, 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 of this bound we can restrict our treatment to the case where $t_p = 0$, i.e., we tolerate an adversarially controlled PKI only if no player is corrupted. This is sufficient, since $t_p > 0$ implies $T < N/2$ — a sub-case for which a protocol has already been given in [FHW04].

For the following construction of a HBC protocol, we will first assume that the PKI is correct. In Sec. 2.3, we then exhibit a generic way of how to fix a possibly adversarially controlled PKI. For our construction, a protocol to satisfy the following definition will be constructed as a building block first.

Definition 3 (Broadcast with extended validity (BCEV) [FHHW03]). *A protocol among N players 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.*
TERMINATION: *All honest players terminate the protocol.* ◇

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.

2.1 An Efficient Protocol for BCEV

Protocol Π_{bcev} is described by the local view of each player P_i (see Figure 1). 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. We construct protocol Π_{bcev} by combining BGP with selective use of signatures: the idea being that, for $t \leq t_\sigma$, BGP will work correctly and dominate the final outcome whereas, for $t_\sigma < t$, at least one forgery would be necessary in order to violate validity of the protocol. The protocol works as follows and is summarized in Figure 1:

- 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 possible input 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, Step 3.

1. P_s: send $(x_s, \sigma_s(x_s))$. [receive (x_i, σ_i)]
2. $\forall \mathsf{P}_i$: 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 correct 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 $T + 2t_\sigma < N$ ($t_\sigma \leq T$) 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 Branch (I) or (II).
- **Broadcast ($t \leq t_\sigma$): Consistency.** As $t \leq t_\sigma < N/3$, BGP is reliable. So, for all honest P_i, P_j BGP results match, implying $S_j^0 = S_i^0$ and $S_j^1 = S_i^1$. Hence, 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 and consistency is guaranteed. Thus, it only remains to consider the case that some honest P_i decides according to Branch (I): Then, $|S_i^{x_i,0}| \geq N - T$ and $|S_i^{1-x_i}| = 0$. We distinguish two cases:
  1. **honest P_j also decides according to (I).** As BGP is reliable ($t_\sigma < N/3$), any honest player P_h succeeds in broadcasting the message he first sent, so $\mathsf{P}_h \in S_i^{x_i,0}$ implies $\mathsf{P}_h \in S_i^{x_i}$. As 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 $S_j^{1-x_i} = S_i^{1-x_i} = \emptyset$. Furthermore, by the same argument as above $|S_j^{x_i}| \geq N - T - t_\sigma > 0$ 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, for honest sender P_s an honest player P_i sees $|S_i^{x_s,0}| \geq N - T$ and $S_i^{1-x_s} = \emptyset$ and thus computes $y_i = x_s$. □

2.2 Achieving HBC from BCEV

We now construct an HBC protocol by combining the BCEV protocol above with the PKI-based BC protocol of [DS82]. Protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$ (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 of [DS82] tolerating any number of corrupted players, relying on a PKI and respective signatures.

– In Step 1 of Protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$, the sender P_s distributes his input in_s using an instance of DS-BC.
– In Step 2 the sender distributes his input in_s using 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.[5]
– 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 these P_j then player 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 Π_{bcev} on sender input in_s.	[receive y_i]	
3.	Multi-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 valid signatures		(I)
	then DS-broadcast the respective signatures and $out_i := v$,	[for each P_k, receive S_i^k]	
	`else` DS-broadcast $\emptyset$.	[for each P_k, receive S_i^k]	
	`If` $\exists$ value v with a set S_i^j of valid signatures on v and $\lvert S_i^j \rvert \geq N - t_\sigma$ then $out_i := v$		(II)
	`else` $out_i := x_i$		(III)
	`fi`		
	`fi`		

Fig. 2. Protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$.

Achieving HBC given a correct PKI means that our protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$ has to be able to tolerate signature forgeries in presence of adversaries corrupting at most $t \leq t_\sigma$ players, while providing BC in presence of adversaries corrupting at most $t \leq T$ players. We now show that protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$ achieves this efficiently for any choice of thresholds t_σ and T where $T + 2t_\sigma < N$.

Lemma 2. *Assuming a correct PKI, protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$ 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 and broadcast is achieved.

[5] We use some unique message tag in order to separate these signatures from those in DS-BC.

• $t_\sigma < t \leq T$**.** In this case we can restrict our attention to adversaries that cannot forge signatures.

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). Then an honest 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 $out_j = out_i$. □

2.3 Fixing the PKI

Previously we assumed that the given PKI was always correct. However, it is our goal to provide a secure BC protocol for any choice of parameters t_σ, t_p, and T where $T + 2t_\sigma < N \ \wedge \ (t_p > 0 \ \Rightarrow \ 2T + t_p < N)$. This protocol must tolerate and adversarially controlled PKI in presence of $t \leq t_p$ corruptions. Our solution for the case $t_p = 0$ must thus be adapted to tolerate an adversarially controlled PKI for the case that $t = 0$ players are corrupted, i.e., for the case that all players are honest. We achieve this by giving a generic construction that transforms a possibly adversarially controlled PKI into a correct one under the condition that all players are honest. The transformation is based on Protocol FGHHS in [FGH$^+$02] for detectable precomputation of a PKI tolerating any number of corrupted players. This protocol sets up a PKI such 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 an honest player accepts the protocol outcome then the PKI is correct.

The transformation works as follows: Let PKI be the given PKI that might be adversarially controlled. Before executing the HBC protocol $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$, an instance of FGHHS is executed resulting in a second PKI instance PKI'.

If one (and thus all) honest player accepts the new PKI' then PKI' is correct as guaranteed by the FGHHS protocol. In this case the players use the new PKI' in the HBC protocol instead of the original PKI. If one (and thus all) honest player rejects the new PKI' then some player must be corrupted as by the guarantee of the FGHHS protocol. In this case we have $t > t_p = 0$ and the protocol is not expected to tolerate an adversarially controlled PKI; and the players can use the original PKI in the HBC protocol.

Denote by Π_{hbc} the protocol that, for $t_p = 0$, runs FGHHS and $\Pi_{\mathsf{hbc}}^{\mathsf{pki}}$, and for $t_p > 0$, runs the BC protocol from [FHW04]. We then arrive at the following

Lemma 3. *Protocol Π_{hbc} efficiently achieves HBC 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. □

Combining this result with the impossibility results of Sec. 4 we get:

Theorem 1. *HBC is achievable (and then efficiently) 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. 6 in Sec. 4. □

3 Hybrid Multi-Party Computation (HMPC)

We now discuss Hybrid Multi-Party Computation (HMPC) in the setting where players are provided with a complete network of secure channels and a possibly adversarially controlled PKI.

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 a threshold t_σ for tolerating adversaries that may forge signatures and a threshold t_p for tolerating an adversarially controlled PKI. Additionally, we define a threshold t_c for tolerating the adversary to be computationally unbounded (given that the PKI is based on unconditionally secure pseudo-signatures). This threshold is independent from the other ones as we will always be able to tolerate $t_c < N/2$ which is optimal by arguments in [RB89,Kil00].

Furthermore, as suggested by the different standard property increments of the literature, we may also define different thresholds for robustness ℓ_r (i.e., full security for $t \leq \ell_r$), fairness ℓ_f (i.e., fair security for $t \leq \ell_f$), and abort L (i.e., abort security for $t \leq L$) — see Sec. 1.1. We refer to these thresholds as *limits*. Note that thresholds refer to adversarial constraints whereas limits refer to the different increments of MPC. In consistency with this separation, limit L has taken the role of the upper threshold T in HBC, $L = T$.

Naturally, we assume that $\ell_r \geq \max(t_p, t_\sigma) = t_\sigma$ since we want to demand full security at least as long as the adversary cannot forge signatures. That is, degradation of security properties shall only be tolerated under the strongest possible adversary.

Note that, although we are dealing with a large number of different threshold parameters for the moment, we will end up at a concise simplification in Sec. 3.2 — without loss of generality. This simplification will allow to fix the single threshold parameter L whereas all other threshold parameters will be able to be maximized independently of each other, i.e., any optimally resilient protocol will be characterized by the single parameter L.

Definition 4 (Hybrid Multi-Party Computation (HMPC)). *Let t_σ, t_p, t_c be thresholds, and ℓ_r, ℓ_f, L be limits, with the additional constraint that $t_p \leq t_\sigma \leq \ell_r \leq \ell_f \leq L$. 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 correct, otherwise it may be controlled by the adversary.*

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

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

In [LRM10], 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, and L. In keeping with the above definition of these limits, the MPC protocol π^ρ_{SA} of [LRM10] is fully secure against $t \leq \ell_r$, fair secure against $t \leq \ell_f$, and abort secure against $t \leq L$ corrupted players.[6] They demonstrate the following bound for the achievability of such MPC:

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

[6] For now, we use the stand-alone secure protocol variant π^ρ_{SA} of the protocol of [LRM10]. The UC setting is discussed in App. A.

This bound is tight as shown in [IKLP06,Kat07,Cle86,Kil00] and discussed in [LRM10]. We will show that HMPC is achievable if and only if both the bounds of Eq. (2) and Thm. 1 are satisfied, i.e.,

Theorem 2 (Bounds for HMPC). *HMPC with thresholds t_σ, t_p, t_c, and limits ℓ_r, ℓ_f, and L, where $t_p \leq t_\sigma \leq \ell_r \leq \ell_f \leq L$, is achievable if and only if*

$$L + 2t_\sigma < N \ \wedge \ (t_p > 0 \ \Rightarrow \ 2L + t_p < N) \qquad \wedge \qquad 2t_c < N \ \wedge \ 2\ell_f < N \ \wedge \ L + \ell_r < N \ . \quad (3)$$

Actually, our impossibility results in Sec. 4 even imply that agreement in MPC comes for free: limiting our requirements to privacy and correctness without agreement allows for no higher threshold L.

3.1 Proof of Thm. 2

Necessity of the right part (Eq. (2)) of Bound (3) directly follows from [LRM10] as their model is strictly more powerful. It remains to demonstrate necessity of the left part of Bound (3). For this, we exhibit a special case of HMPC that is not achievable beyond. Note that HBC itself cannot be such a candidate instance since HMPC does not imply HBC (as HMPC does not require robustness). However, if we augment HBC with the possibility of such non-robustness, we arrive at exactly such an instance: hybrid broadcast with abort, HBCA for short. That is, HBCA is the same as HBC except for the additional limit $\ell_r \geq t_\sigma$ and the additional property that robustness is only required if $t \leq \ell_r$. In Sec. 4, impossibility of HBCA beyond $L + 2t_\sigma < N$ and $t_p > 0 \ \Rightarrow \ 2L + t_p < N$ is demonstrated. This finishes the necessity argument of the proof.

We now argue sufficiency by exhibiting an HMPC protocol that achieves the bounds above. We combine the MPC protocol π^ρ_{SA} of [LRM10] 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.[6] The MPC protocol π^ρ_{SA} of [LRM10] is fully secure for $t \leq \ell_r$, fair secure for $t \leq \ell_f$, and abort secure for $t \leq L$ corrupted players. The protocol Π_{hbc} is secure under the bounds of Thm. 1.

As the MPC protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ exhibits the same security properties as protocol π^ρ_{SA} when run with BC channels, 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 be thresholds and ℓ_r, ℓ_f, L be limits where $t_p \leq t_\sigma \leq \ell_r \leq \ell_f \leq L$ as in Thm. 2, i.e.,*

$$L + 2t_\sigma < N \ \wedge \ (t_p > 0 \ \Rightarrow \ 2L + t_p < N) \qquad \wedge \qquad 2t_c < N \ \wedge \ 2\ell_f < N \ \wedge \ L + \ell_r < N \ .$$

Then Protocol $\pi^\rho_{\mathsf{SA}} \circ \Pi_{\mathsf{hbc}}$ achieves hybrid multi-party computation (HMPC) *secure against a static active adversary corrupting up to t of the players.*

Proof. The theorem follows from the discussion above. □

This completes the proof of Thm. 2. □

A discussion of HMPC in the UC setting including a UC restatement of Thm. 2 can be found in App. A.

3.2 Simplification

We now simplify Thm. 3 by expressing all bounds implicitly by the parameter L. Parameter t_c is independent of all other parameters and we may already fix it to its maximal possible value $t_c = \lfloor \frac{N-1}{2} \rfloor$. We distinguish the cases $L < N/3$, $N/3 \leq L < N/2$, and $L \geq N/2$, where the first case is covered by [BGW88,CCD88].

For $N/3 \leq T < N/2$ we arrive at HMPC with the properties in [FHW04]. Note that the constraint $N/3 \leq T$ together with the implicit bounds in the following proposition automatically imply that $t_p \leq t_\sigma$ as required by the model.

Corollary 1 ([FHW04] $\mathbf{N/3 \leq L < N/2}$). *Let a PKI and a complete network of secure channels be given and let L be fixed such that $N/3 \leq L < N/2$. And let $t \leq L$ be the number of corrupted players. Then fully secure HMPC is achievable exactly under the following adversarial constraints*

1. *the adversary is allowed to be computationally unbounded, and*
2. *if $t < N - 2L$ then the adversary is allowed to control the PKI, and*
3. *if $t < \frac{N-L}{2}$ then the adversary is allowed to forge signatures.*

In particular, these bounds are independent in the sense that the lowering of one threshold does not allow for an increase of any other threshold.

For $L \geq N/2$ we obtain new results that go beyond those of [FHW04]. Let us fix L anywhere such that $L \geq N/2$. Our bound then directly implies $t_p = 0$. Without loss, we can now independently maximize the thresholds t_σ, ℓ_r, and ℓ_f, with respect to L without violating that $t_\sigma \leq \ell_r \leq \ell_f$.

Corollary 2 ($\mathbf{L \geq N/2}$). *Let a PKI and a complete network of secure channels be given and let L be fixed such that $L \geq N/2$. And let $t \leq L$ be the number of corrupted players. Then HMPC is achievable exactly under the following adversarial constraints*

1. *if $t < \frac{N}{2}$ the adversary is allowed to be computationally unbounded, and*
2. *if $t = 0$ the adversary is allowed to control the PKI, and*
3. *if $t < \frac{N-L}{2}$ the adversary is allowed to forge signatures,*

and providing the security properties

1. *full security if $t < N - L$, and*
2. *fair security if $t < \frac{N}{2}$, and*
3. *abort security.*

In particular, these bounds are independent in the sense that the lowering of one threshold does not allow for an increase of any other threshold.

4 Impossibility

We demonstrate impossibility of HBC and HMPC beyond the bounds of Thms. 1 and 2 by showing that HBCA, i.e., hybrid broadcast with abort, is not achievable. HBCA is the same as HBC except that robustness is only required if $t \leq t_\sigma$.

Definition 5 (HBCA). *A protocol with parameters t_p, t_σ, and T, $t_p \leq t_\sigma \leq T$, among a player set $\mathcal{P}$, $|\mathcal{P}| = N$, where a player $\mathsf{P}_s \in \mathcal{P}$ (the sender) inputs a value $x_s \neq \perp$ and every player $\mathsf{P}_i \in \mathcal{P}$ outputs a value y_i achieves* hybrid broadcast with abort (HBCA) *if, under the condition that*

- *the PKI may be controlled by the adversary if $t \leq t_p$, and*
- *the adversary may forge signatures if $t \leq t_\sigma$,*

the following holds:

- *if $t \leq t_\sigma$ then the protocol achieves broadcast, and*
- *if $t \leq T$ then the protocol either achieves broadcast or all honest players terminate the protocol with output $\perp$.* ◇

Impossibility of HBCA 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 HBCA is impossible if $t_p = 0$, $t_\sigma > 0$, and $T + 2t_\sigma \geq N$.
2. Second, we show that HBCA is impossible if $t_p > 0$ and $2T + t_p \geq N$.

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

For the sake of contradiction, assume a protocol that achieves HBCA under these conditions among a player set $\mathcal{P}$, $|\mathcal{P}| = N \geq 3$. We can partition the players in to three sets $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_2$, with cardinalities $|\mathcal{P}_0| \leq T$ and $|\mathcal{P}_1|, |\mathcal{P}_2| \leq t_\sigma$ where the sender P_s is in $\mathcal{P}_0$. Let P'_i be a copy of player $\mathsf{P}_i \in \mathcal{P}_0$ and $\mathcal{P}'_0 = \{\mathsf{P}'_i | \mathsf{P}_i \in \mathcal{P}_0\}$ 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 $\mathcal{P}' = \mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}'_0$ in a certain way and let the protocol run.

The players are connected in the following way — see Figure 3. Exactly all pairs in $(\mathcal{P}_0 \cup \mathcal{P}_1) \times (\mathcal{P}_0 \cup \mathcal{P}_1)$, $(\mathcal{P}_1 \cup \mathcal{P}_2) \times (\mathcal{P}_1 \cup \mathcal{P}_2)$, and $(\mathcal{P}_2 \cup \mathcal{P}'_0) \times (\mathcal{P}_2 \cup \mathcal{P}'_0)$ are connected by pairwise channels meaning that a message that normally would be sent from $\mathsf{P}_i \in \mathcal{P}_2$ to $\mathsf{P}_j \in \mathcal{P}_0$ is sent from P_i to $\mathsf{P}'_j \in \mathcal{P}'_0$ instead, and that P'_j communicates with the players in $\mathcal{P}_2 \cup \mathcal{P}'_0$ as it would with the players in $\mathcal{P}_2 \cup \mathcal{P}_0$ under normal conditions. Note that no further connections exist.

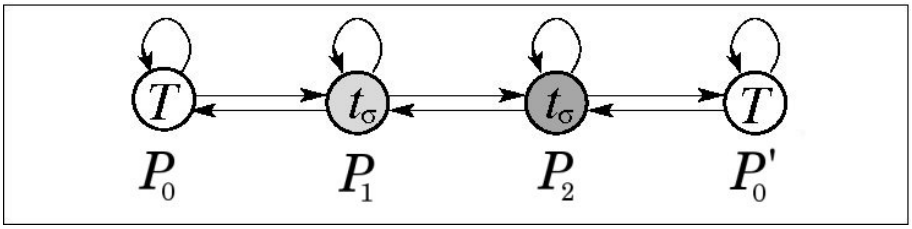

Fig. 3. Simulated system by the adversary for the case $t_\sigma > 0$

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 $\mathcal{P}_0 \cup \mathcal{P}_1$, $\mathcal{P}_1 \cup \mathcal{P}_2$, and $\mathcal{P}_2 \cup \mathcal{P}'_0$, are indistinguishable from their joint view in a protocol under normal conditions where the adversary corrupts the remaining players.

Joint view of $\mathcal{P}_0 \cup \mathcal{P}_1$ with $x_s = 0$. By corrupting all players in $\mathcal{P}_2$ in the original system the adversary simulates all players in $\mathcal{P}_2 \cup \mathcal{P}'_0$ of the new system. Since $|\mathcal{P}_2| \leq t_\sigma$, the adversary can forge all signatures by players in $\mathcal{P}'_0$ required for the simulation. Thus the joint view of the players in $\mathcal{P}_0 \cup \mathcal{P}_1$ in the original system is exactly the same as their view in the new system.

Joint view of $\mathcal{P}_2 \cup \mathcal{P}'_0$ with $x'_s = 1$. By symmetry, this case follows from the above paragraph.

Joint view of $\mathcal{P}_1 \cup \mathcal{P}_2$. By corrupting all players in $\mathcal{P}_0$ in the original system the adversary can simulate all players in $\mathcal{P}_0 \cup \mathcal{P}'_0$ of the new system. Note that, by corrupting the players in $\mathcal{P}_0$, 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 $\mathcal{P}_1 \cup \mathcal{P}_2$ in the original system is exactly the same as their view in the new system.

Contradiction. The assumption that the given protocol achieves HBCA implies that the players $\mathsf{P}_i \in \mathcal{P}_0 \cup \mathcal{P}_1$ must agree on $y_i = x_s = 0$ since the adversary corrupts the at most t_σ players in $\mathcal{P}_2$. By the same argument, the players $\mathsf{P}_j \in \mathcal{P}_2 \cup \mathcal{P}'_0$ must agree on $y_j = x'_s = 1$. However, this implies that the players in $\mathcal{P}_1$ and the players in $\mathcal{P}_2$ disagree on their outputs in contradiction to the definition of HBCA. This implies that HBCA is not achievable under these conditions.

Lemma 4. *If $t_\sigma > 0$ then HBCA is not achievable if $T + 2t_\sigma \geq N$.*

Proof. The proof follows from the above discussion. □

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

We proceed in the same way as in the previous section. We can partition the players in to three sets $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_2$, with cardinalities $|\mathcal{P}_0| \leq t_p$ and $|\mathcal{P}_1|, |\mathcal{P}_2| \leq T$ where the sender P_s is in $\mathcal{P}_0$. Let P'_i, $\mathcal{P}'_0$, and $\mathcal{P}'$ be defined as in the previous section.

The players are connected in the following way — see Figure 4. Exactly all pairs in $(\mathcal{P}_0 \cup \mathcal{P}_1) \times (\mathcal{P}_0 \cup \mathcal{P}_1)$, $(\mathcal{P}_1 \cup \mathcal{P}_2) \times (\mathcal{P}_1 \cup \mathcal{P}_2)$, and $(\mathcal{P}_2 \cup \mathcal{P}'_0) \times (\mathcal{P}_2 \cup \mathcal{P}'_0)$ are connected as in the previous section. Additionally, for all players $\mathsf{P}'_i \in \mathcal{P}'_0$ 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 $\mathsf{P}_j \in \mathcal{P}_2 \cup \mathcal{P}'_0$ P_j's copy of PK_i is replaced by PK'_i.

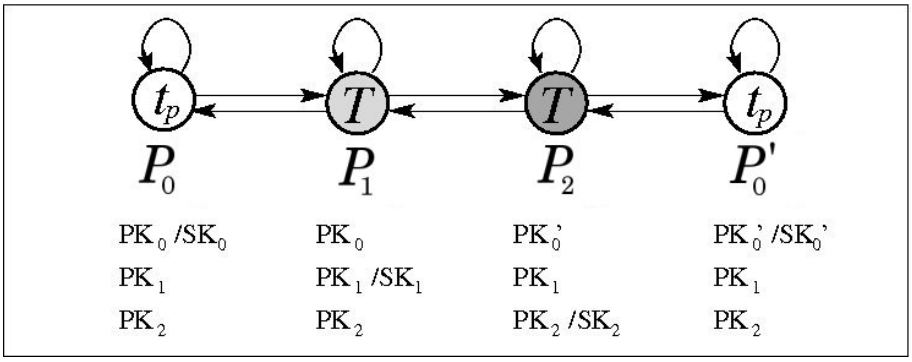

Fig. 4. Simulated system by the adversary for the case $t_p > 0$

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 $\mathcal{P}_0 \cup \mathcal{P}_1$, $\mathcal{P}_1 \cup \mathcal{P}_2$, and $\mathcal{P}_2 \cup \mathcal{P}'_0$, are indistinguishable from their joint view in a protocol under normal conditions where the adversary corrupts the remaining players.

Joint view of $\mathcal{P}_0 \cup \mathcal{P}_1$ with $x_s = 0$. By corrupting all players in $\mathcal{P}_2$ in the original system the adversary simulates all players in $\mathcal{P}_2 \cup \mathcal{P}'_0$ of the new system. For all $\mathsf{P}_i \in \mathcal{P}'_0$ 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 $\mathsf{P}_j \in \mathcal{P}_2 \cup \mathcal{P}'_0$, overwrites P_j's copy of P_i's public key. The PKI among the players in $\mathcal{P}_0 \cup \mathcal{P}_1$ is still fully correct and thus the joint view of the players in $\mathcal{P}_0 \cup \mathcal{P}_1$ in the original system is exactly the same as their view in the new system.

Joint view of $\mathcal{P}_2 \cup \mathcal{P}'_0$ with $x'_s = 1$. By symmetry, this case follows from the above case.

Joint view of $\mathcal{P}_1$ and $\mathcal{P}_2$. Since $|\mathcal{P}_0| \leq t_p$ the adversary can control the PKI and may distribute keys according to Figure 4. By corrupting all players in $\mathcal{P}_0$ in the original system the adversary can now simulate all players in $\mathcal{P}_0 \cup \mathcal{P}'_0$ of the new system. Thus the joint view of the players in $\mathcal{P}_1 \cup \mathcal{P}_2$ in the original system is exactly the same as their view in the new system.

Contradiction. Assuming the protocol to achieve HBCA now implies that the players $\mathsf{P}_i \in \mathcal{P}_0 \cup \mathcal{P}_1$ must output $y_i \in \{x_s, \perp\} = \{0, \perp\}$ since at most T players are corrupted. By the same argumentation, the players $\mathsf{P}_j \in \mathcal{P}_2 \cup \mathcal{P}'_0$ must output $y_j \in \{x'_s, \perp\} = \{1, \perp\}$. However, this means that the players in $\mathcal{P}_1$ and the players in $\mathcal{P}_2$ either disagree on their outputs or jointly output $\perp$, which is, under corruption of $t \leq t_p$ players, in contradiction to the definition of HBCA. This implies that HBCA is not achievable under these conditions.

Lemma 5. *If $t_p > 0$ then HBCA is not achievable if $2T + t_p \geq N$.*

Proof. The proof follows from the above discussion. □

Lemma 6. *HBC, HBCA, and HMPC are not not achievable if $t_\sigma > 0$ and $T + 2t_\sigma \geq N$, or if $t_p > 0$ and $2T + t_p \geq N$ (for HMPC, replace T with L).*

Proof. The lemma follows from Lem. 4, Lem. 5, the fact that HBC implies HBCA, and the fact that HBCA is an instance of HMPC. □

Note that the impossibility result for HMPC even applies if we merely demand privacy and correctness without robustness, fairness, or agreement for $t > t_\sigma$, as private and correct MPC already implies BCA, and thus such HMPC implies HBCA. As such our impossibility result implies that agreement in HMPC comes for free: limiting our requirements to privacy and correctness without agreement allows for no higher threshold L.

5 Conclusions

We presented tight bounds and optimal protocols for hybrid broadcast (HBC) and hybrid multi-party computation (HMPC) for a setting where a (possibly adversarially controlled) PKI and a complete network of secure channels, but no broadcast channels, are provided. This can be seen as extending the work of [FHW04] to the setting where robustness is not always required, or as extending the work of [LRM10] to the setting where a possibly unreliable PKI is given instead of a reliable BC channel.

References

[Bea89] Donald Beaver. Multiparty protocols tolerating half faulty processors. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 560–572. Springer, 1989.

[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.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10. ACM, 1988.

[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 Bob Werner, editor, *IEEE Symposium on Symposium on Foundations of Computer Science (FOCS) 2001*, pages 136–147, Los Alamitos, CA, October 14–17 2001. IEEE Computer Society.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In *ACM Symposium on the Theory of Computing (STOC) 1988*, pages 11–19. ACM, 1988.

[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.

[CS99] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *6th ACM Conference on Computer and Communications Security (CCS '99)*, pages 46–51, New York, 1999. ACM Press.

[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.

[GGBS10] Anuj Gupta, Prasant Gopal, Piyush Bansal, and Kannan Srinathan. Authenticated byzantine generals in dual failure model. In Krishna Kant, Sriram V. Pemmaraju, Krishna M. Sivalingam, and Jie Wu, editors, *ICDCN*, volume 5935 of *Lecture Notes in Computer Science*, pages 79–91. Springer, 2010.

[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.

[GKKY09] S. Dov Gordon, Jonathan Katz, Ranjit Kumaresan, and Arkady Yerukhimovich. Authenticated broadcast with a partially compromised public-key infrastructure. Cryptology ePrint Archive, Report 2009/410, 2009. `http://eprint.iacr.org/`.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *ACM Symposium on Theory of Computing*, pages 218–229, 1987.

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

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, 2001.

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

[HZ10] Martin Hirt and Vassilis Zikas. Adaptively secure broadcast. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 466–485. Springer, 2010.

[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.

[LRM10] Christoph Lucas, Dominik Raub, and Ueli M. Maurer. Hybrid-secure mpc: trading information-theoretic robustness for computational privacy. In Andréa W. Richa and Rachid Guerraoui, editors, *PODC*, pages 219–228. ACM, 2010. Full version available at `eprint.iacr.org/2009/009`.

[PW96] Birgit Pfitzmann and Michael Waidner. Information-theoretic pseudosignatures and byzantine agreement for $t >= n/3$. Research Report RZ 2882 (#90830), IBM Research, November 1996.

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

[RB89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. 21st annual ACM symposium on theory of computing (STOC '89)*, pages 73–85, 1989.

[Yao82] Andrew C. Yao. Protocols for secure computations (extended abstract). In *IEEE Symposium on Symposium on Foundations of Computer Science (FOCS) 1982*, pages 160–164, Chicago, Illinois, 1982. IEEE.

A UC Security

In this section we place our result into the context of universally composable (UC) security.

A.1 Security Definitions and Notations

We follow the Universal Composability (UC) paradigm [PW00,Can01,BPW04][7], which defines a simulation-based security model. The security of a protocol (the real world) is defined with respect to a *Trusted Third Party* or *Ideal Functionality* F that correctly performs all desired computations (the ideal world). Informally, a protocol π is secure if whatever an adversary can achieve in the real world could also be achieved in the ideal world.

More precisely, let $\mathcal{P} = \{\mathsf{P}_1, \ldots, \mathsf{P}_N\}$ be the set of players, and define $[N] := \{1, \ldots, N\}$. We only consider static corruptions and use $\mathcal{H} \subseteq \mathcal{P}$ to denote set of honest players, and $\mathcal{A} = \mathcal{P} \setminus \mathcal{H}$ to denote the set of corrupted players. In the *real world*, there is a given set of resources R (e.g., secure channels, broadcast channels) to which for each honest player $\mathsf{P}_i \in \mathcal{H}$ a protocol machine π_i is connected. Corrupted players access the resources directly. This real world is denoted by $\pi_{\mathcal{H}}(\mathsf{R})$. In [Can01], resources are modeled as ideal functionalities available in a hybrid model . The *ideal world* consists of the ideal functionality F and an ideal adversary (or simulator) S connected to F via the interfaces of the corrupted players $\mathcal{A}$. This ideal world is denoted by $\mathsf{S}_{\mathcal{A}}(\mathsf{F})$.

A protocol π securely implements a functionality F if, for every possible set $\mathcal{A}$ of corrupted players, there is a simulator S such that no distinguisher D can tell the real world and the ideal world apart.[8] For this purpose, the distinguisher directly interacts either with the real or with the ideal world, by connecting to all open interfaces, and then outputs a decision bit. This interaction is denoted by $\mathsf{D}(X)$, where $X \in \{\pi_{\mathcal{H}}(\mathsf{R}), \mathsf{S}_{\mathcal{A}}(\mathsf{F})\}$.

In [Can01], all protocol machines, simulators, ideal functionalities, and distinguishers are modeled as Interactive Turing Machines (ITM). We define Σ^{all} as the set of *all* ITMs, and Σ^{eff} as the set of *polynomially bounded* ITMs. In this paper, however, ITMs are specified on a higher level of abstraction.

Definition 6 (Universally Composable (UC) Security). *A protocol* π UC securely *implements an ideal functionality* F *if*

$$\forall \mathcal{A}\ \exists \mathsf{S}_{\mathcal{A}} \in \Sigma^{\mathsf{eff}}\ \forall \mathsf{D} \in \Sigma^{\mathsf{eff/all}} :\ |Pr[\mathsf{D}(\mathsf{S}_{\mathcal{A}}(\mathsf{F})) = 1] - Pr[\mathsf{D}(\pi_{\mathcal{H}}(\mathsf{R})) = 1]| \leq \varepsilon(\kappa),$$

where $\varepsilon(\kappa)$ *denotes a negligible function in the security parameter* κ*. For* $\mathsf{D} \in \Sigma^{\mathsf{eff}}$*, the security is* computational *(CO). For* $\mathsf{D} \in \Sigma^{\mathsf{all}}$*, the security is* information-theoretic *(IT).*

Simulators must be efficient not only in the CO, but also in the IT setting, since otherwise, IT security does not imply CO security. We formalize hybrid security using ideal functionalities that are aware of both the set of corrupted players and the computational power of the adversary. In other words, the behavior of the functionality, and hence the security guarantees, varies depending on both parameters. A protocol π UC securely implements an ideal functionality F with hybrid-security if π securely implements F in both the CO and the IT setting. Note that in contrast to [Can01], we use a synchronous communication model with static corruption. As resources R we will, unless otherwise stated, assume a complete network Net^N of synchronous secure channels. For MPC we will also need a common reference string (CRS) CRS to avoid the impossibility results of [Can01,CF01].

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

[8] In this model, the adversary is thought of as being part of the distinguisher. Canetti [Can01] shows that this model without adversary is essentially equivalent to a model with adversary, since the security definition quantifies over all distinguishers.

In the UC setting, a strong composition theorem can be proven [Can01,BPW04]. This theorem 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.[9]

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 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.[10]

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

A.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.

We are interested in implementing BC and MPC in a hybrid-setting where a PKI is provided, but where the adversary may control the PKI if he corrupts less that $t \leq t_p$ players, or where the adversary may be able to forge signatures if he corrupts less that $t \leq t_\sigma$ players.

We model this setting by providing an unreliable PKI resource $\mathsf{PKI}_{t_p,t_\sigma}$ that models the capabilities of the adversary. For simplicity let an unconditionally secure pseudo-signature scheme (e.g. [PW96]) be given. We could also use a signature scheme existentially unforgeable under adaptive chosen-message attacks (UF-CMA) (e.g. [CS99]), but then unbounded adversaries can forge signatures, which complicates treatment. Functionality $\mathsf{PKI}_{t_p,t_\sigma}$ then operates as follows:

– In case of $t \leq t_p$ corruptions, functionality $\mathsf{PKI}_{t_p,t_\sigma}$ turns control over to the adversary, so the adversary may fix all public and private keys arbitrarily.

[9] 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. 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.

[10] 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.

- In case of $t_p < t \leq t_\sigma$ corruptions, functionality $\mathsf{PKI}_{t_p,t_\sigma}$ independently and honestly generates keys for each player $\mathsf{P}_i \in \mathcal{P}$ according to the prescribed signature scheme and distributes them. Then functionality $\mathsf{PKI}_{t_p,t_\sigma}$ reveals all signing keys to the adversary, to model that the adversary may forge signatures.
- In case of $t > \max(t_\sigma, t_p)$ corruptions, functionality $\mathsf{PKI}_{t_p,t_\sigma}$ independently and honestly generates keys for each player $\mathsf{P}_i \in \mathcal{P}$ according to the prescribed signature scheme and distributes them.

We can now restate the security of our HBC protocol Π_{hbc} as claimed in Lem. 3 in the UC setting as follows

Lemma 8 (UC security of HBC). *Protocol Π_{hbc} efficiently and UC securely implements functionality* bc *in the IT setting*[11] *from a PKI $\mathsf{PKI}_{t_p,t_\sigma}$ and a network Net^N as resources in presence of at most $t \leq T$ actively corrupted players whenever*

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

Proof of Lem. 8 Efficiency easily follows by inspection of the protocols. We show that the protocol Π_{hbc} indeed implements functionality bc whenever at most $t \leq T$ players are corrupted and $T+2t_\sigma < N \wedge (t_p > 0 \ \Rightarrow \ 2T + t_p < N)$ by providing an appropriate simulator S^{bc}:

The simulator S^{bc} connects to the interfaces of corrupted players to functionality bc. Simulator S^{bc} internally emulates the protocol machines Π^i_{hbc} for the honest players P_i and an instance of the PKI $\mathsf{PKI}_{t_p,t_\sigma}$. The connections to corrupted players are exposed to 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 bc 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 bc. This amounts to nothing else then demanding consistency and validity as proven above.

A.3 UC Secure HMPC

We now translate Sec. 3 to the UC setting. First, we formalize HMPC by providing an ideal functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$. This functionality evaluates an arbitrary N-player functionality F with the HMPC properties:

Definition 7 (Functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$). *Given an arbitrary N-player functionality F, functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$ behaves as follows:*

1. *If $t > t_c$ and the adversary is computationally unbounded, or*
2. *if $t > L$*

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

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

[11] If we want to use a computational PKI, we have to restrict our attention to distinguishers that cannot forge signatures.

Consider the protocol $\pi^{\rho} \circ \Pi_{\mathsf{hbc}}$ obtained from the MPC protocol π^{ρ} of [LRM10] by using out HBC protocol Π_{hbc} for broadcasts. As for the stand-alone setting in Sec. 3 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. (3).

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 in the following assume as resources R a common reference string CRS and a complete network Net^N of synchronous secure channels and an unreliable PKI $\mathsf{PKI}_{t_p,t_\sigma}$.A correctly chosen CRS is a prerequisite to the security of the protocols from [LRM10].[12]

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 $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$ from an unreliable PKI $\mathsf{PKI}_{t_p,t_\sigma}$, a complete network of synchronous secure channels Net^N, and a CRS for any choice of thresholds respecting*

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

in presence of an active, static adversary.

Proof. The proof of Thm. 4 is almost trivial.

By Lem. 8 the BC protocol Π_{hbc} implements functionality bc under the bounds of Lem. 3. Now, by the choice of bounds in the definition of functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$, we then find: In any setting where functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$ does not turn over control to the adversary, the BC protocol Π_{hbc} implements functionality bc. According to the UC Theorem it is hence sufficient to prove that protocol $\pi^{\rho} \circ \mathsf{bc}$ implements functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$. But on a plain bc, protocol π^{ρ} provides precisely the guarantees made by functionality $\mathsf{F}^{\mathsf{hyb}}_{t_c,\ell_r,\ell_f,L}$ (also see [LRM10]). □

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

[12] As noted in [LRM10], 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.