Closing the Efficiency Gap between Synchronous and Network-Agnostic Consensus

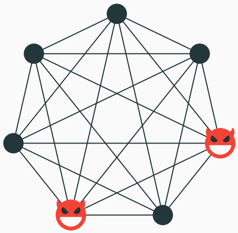
Giovanni Deligios Mose Mizrahi Erbes

ETH Zurich

Consensus (Byzantine Agreement)

Parties $P_1, P_2, \ldots, P_n$ with ℓ-bit inputs.

Up to t of the parties are byzantine.



Consistency

The parties agree on an output.

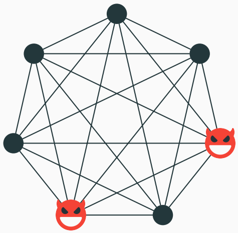
Validity

common input $\boldsymbol{m} \implies$ output $\boldsymbol{m}$

Consensus (Byzantine Agreement)

Parties $P_1, P_2, \dots, P_n$ with ℓ-bit inputs.

Up to t of the parties are byzantine.



Consistency

The parties agree on an output.

Validity

common input $\boldsymbol{m} \implies$ output $\boldsymbol{m}$

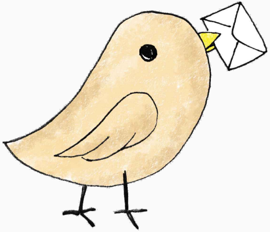
Intrusion Tolerance

The common output is either an honest input, or a special value $\perp$.

Network-Agnostic Setting

Synchronous Setting: Messages arrive after Δ time, clocks synchronized. Security possible with setup when $t < \frac{n}{2}$ [8].

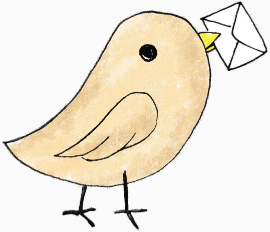
Asynchronous Setting: Messages arrive after arbitrary delays, clocks not synchronized. Security requires $t < \frac{n}{3}$ [15].



Network-Agnostic Setting

Synchronous Setting: Messages arrive after Δ time, clocks synchronized. Security possible with setup when $t < \frac{n}{2}$ [8].

Asynchronous Setting: Messages arrive after arbitrary delays, clocks not synchronized. Security requires $t < \frac{n}{3}$ [15].



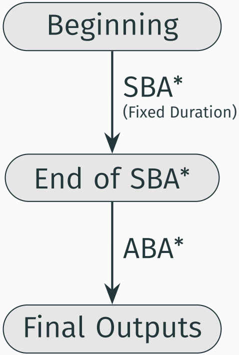
Network-Agnostic Setting (Blum, Katz, Loss [2])

Network synchronous ($\leq t_s$ corruptions), or asynchronous ($\leq t_a$ corruptions). The parties don't know if the network is synchronous or not.

Consensus when $t_a \leq t_s$ is possible iff $2t_s + t_a < n$ [2].

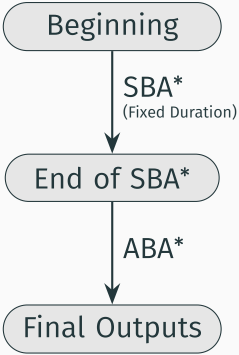
Fallback Compilation

[2] Blum, Katz and Loss compile a synchronous consensus protocol SBA* and an asynchronous consensus protocol ABA*.



Fallback Compilation

[2] Blum, Katz and Loss compile a synchronous consensus protocol SBA* and an asynchronous consensus protocol ABA*.



★ SBA* achieves validity against t_a corruptions, even if the network is asynchronous.

★ ABA* achieves validity against t_s corruptions when the network is synchronous.

Previous Works

2019: Blum, Katz and Loss introduced the setting. Their ABA* required unique threshold signatures for a common coin [2].

- **Round complexity:** $\Omega(n)$

Previous Works

2019: Blum, Katz and Loss introduced the setting. Their ABA* required unique threshold signatures for a common coin [2].

- **Round complexity:** $\Omega(n)$

2021: Deligios, Hirt and Liu-Zhang designed a more round-efficient SBA* [7].

- **Round complexity:** $\mathcal{O}(\lambda)$ for the statistical error probability $2^{-\lambda}$.

Previous Works

2019: Blum, Katz and Loss introduced the setting. Their ABA* required unique threshold signatures for a common coin [2].

- **Round complexity:** $\Omega(n)$

2021: Deligios, Hirt and Liu-Zhang designed a more round-efficient SBA* [7].

- **Round complexity:** $\mathcal{O}(\lambda)$ for the statistical error probability $2^{-\lambda}$.

2023: Bacho, Collins, Liu-Zhang and Loss designed a new ABA* which works with a bulletin-PKI setup, supports ℓ-bit inputs, and has intrusion tolerance [1].

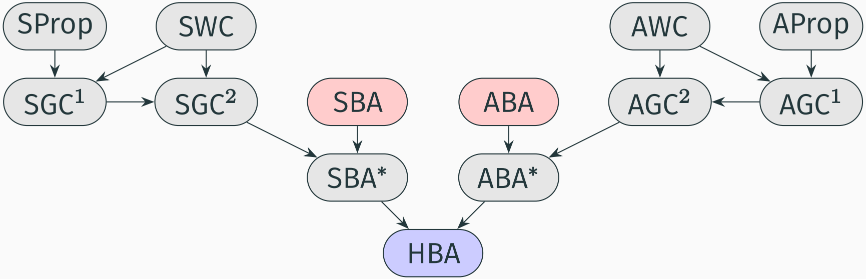
- **Communication complexity:** $\mathcal{O}(n^3\kappa + \ell n^3)$.

The Generic Approach

2022: Ghinea, Goyal and Liu-Zhang designed a λ-round SBA with a statistical error probability $\lambda^{-\Omega(\lambda)}$ when $2t_s \leq (1-\varepsilon)n$ [12]. How do we get this for SBA*?

The Generic Approach

2022: Ghinea, Goyal and Liu-Zhang designed a λ-round SBA with a statistical error probability $\lambda^{-\Omega(\lambda)}$ when $2t_s \leq (1-\varepsilon)n$ [12]. How do we get this for SBA*?



We can compile **any** fixed-duration SBA and **any** ABA.

Overhead: 13 or 16 rounds when the network is synchronous.

Example – Asynchronous 2-Graded Consensus

Assume $t_a \leq t_s$ and $2t_s + t_a < n$.

Asynchronous 2-Graded Consensus

Inputs: $m_i \in \{0,1\}^\ell$ **Outputs:** $(y_i, g_i) \in (\{0,1\}^\ell \cup \{\bot\}) \times \{0,1,2\}$

- **t_s-intrusion tolerance:** If no party has an input m, then no party P_i obtains $y_i = m$.
- **6-round t_s-validity with liveness:** If the parties run forever with a common input m, then they output $(m, 2)$, and do so within 6Δ time if the network is synchronous.

Example – Asynchronous 2-Graded Consensus

Assume $t_a \leq t_s$ and $2t_s + t_a < n$.

Asynchronous 2-Graded Consensus

Inputs: $m_i \in \{0,1\}^\ell$ **Outputs:** $(y_i, g_i) \in (\{0,1\}^\ell \cup \{\bot\}) \times \{0,1,2\}$

- $\boldsymbol{t_s}$**-intrusion tolerance:** If no party has an input m, then no party P_i obtains $y_i = m$.
- **6-round** $\boldsymbol{t_s}$**-validity with liveness:** If the parties run forever with a common input m, then they output $(m, 2)$, and do so within 6Δ time if the network is synchronous.
- $\boldsymbol{t_a}$**-consistency:** For all P_i and P_j, it holds that $|g_i - g_j| \leq 1$ and $g_i \geq 1 \implies y_i = y_j$.
- $\boldsymbol{t_a}$**-liveness:** If the parties all acquire inputs and run forever, then they all output.

Example – Asynchronous 2-Graded Consensus

Assume $t_a \leq t_s$ and $2t_s + t_a < n$.

Asynchronous 2-Graded Consensus

Inputs: $m_i \in \{0,1\}^\ell$ **Outputs:** $(y_i, g_i) \in (\{0,1\}^\ell \cup \{\perp\}) \times \{0,1,2\}$

- **t_s-intrusion tolerance:** If no party has an input m, then no party P_i obtains $y_i = m$.
- **6-round t_s-validity with liveness:** If the parties run forever with a common input m, then they output $(m, 2)$, and do so within 6Δ time if the network is synchronous.
- **t_a-consistency:** For all P_i and P_j, it holds that $|g_i - g_j| \leq 1$ and $g_i \geq 1 \implies y_i = y_j$.
- **t_a-liveness:** If the parties all acquire inputs and run forever, then they all output.

Complexity: $\mathcal{O}(n^2)$ messages, $\mathcal{O}(\ell n^2)$ bits

Asynchronous Consensus – Termination

Old Way [2, 7, 1]

Sign your ABA* output and multicast it.

A $(t_s + 1)$-certificate on y proves y is the correct output. Upon having one, multicast it, output y and terminate.

Termination against t_s corruptions.

Asynchronous Consensus – Termination

Old Way [2, 7, 1]

Sign your ABA* output and multicast it.

A $(t_s + 1)$-certificate on y proves y is the correct output. Upon having one, multicast it, output y and terminate.

Termination against t_s corruptions.

New Way – Bracha's Broadcast Style [3]

Multicast your ABA* output, unsigned.

Upon receiving y from $t_s + 1$ parties, multicast y.

Upon receiving y from $n - t_s$ parties, output y and terminate.

Termination against t_a corruptions.

Problem: We need termination against t_s corruptions in synchronous networks.

Asynchronous Consensus – Termination

Old Way [2, 7, 1]

Sign your ABA* output and multicast it.

A $(t_s + 1)$-certificate on y proves y is the correct output. Upon having one, multicast it, output y and terminate.

Termination against t_s corruptions.

New Way – Bracha's Broadcast Style [3]

Multicast your ABA* output, unsigned.

Upon receiving y from $t_s + 1$ parties, multicast y.

Upon receiving y from $n - t_s$ parties, output y and terminate.

Termination against t_a corruptions.

Problem: We need termination against t_s corruptions in synchronous networks.

Solution: In synchrony, everyone outputs by some time T. Don't terminate earlier.

Bonus Efficiency: Don't send ABA messages before the time $T + \Delta$.

Complexity Summary

Adapting techniques by Momose and Ren [13] and using their SBA to obtain SBA*, we achieve (with no $\mathsf{CC}_{\mathsf{ABA}}$ in synchrony):

Resilience	Setup	Communication Complexity
$2t_s + t_a < n$	Bulletin-PKI	$\mathcal{O}(\mathsf{CC}_{\mathsf{ABA}} + n^3\kappa + \ell n^2)$
$2t_s + t_a < n$	Threshold Signatures	$\mathcal{O}(\mathsf{CC}_{\mathsf{ABA}} + n^2\kappa + \ell n^2)$
$2t_s + t_a < n$, $2t_s \leq (1-\varepsilon)n$	Bulletin-PKI	$\mathcal{O}(\mathsf{CC}_{\mathsf{ABA}} + n^2\kappa + \ell n^2)$

Complexity Summary

Adapting techniques by Momose and Ren [13] and using their SBA to obtain SBA*, we achieve (with no $\mathsf{CC}_{\mathsf{ABA}}$ in synchrony):

Resilience	Setup	Communication Complexity
$2t_s + t_a < n$	Bulletin-PKI	$\mathcal{O}(\mathsf{CC}_{\mathsf{ABA}} + n^3\kappa + \ell n^2)$
$2t_s + t_a < n$	Threshold Signatures	$\mathcal{O}(\mathsf{CC}_{\mathsf{ABA}} + n^2\kappa + \ell n^2)$
$2t_s + t_a < n$, $2t_s \leq (1-\varepsilon)n$	Bulletin-PKI	$\mathcal{O}(\mathsf{CC}_{\mathsf{ABA}} + n^2\kappa + \ell n^2)$

Unique threshold signatures: $\mathsf{CC}_{\mathsf{ABA}} = \mathcal{O}(n^2\kappa)$ [4].

Bulletin-PKI and CRS, or no setup but static security: $\mathsf{CC}_{\mathsf{ABA}} = \mathcal{O}(n^3\kappa)$ [11, 6].

Adaptive security without setup: $\mathsf{CC}_{\mathsf{ABA}} = \mathcal{O}(n^3\kappa \log n)$ [10].

Complexity Summary – Extended

We can reduce the $\mathcal{O}(\ell n^2)$ term to $\mathcal{O}(\ell n)$ with extension protocols.

Thanks to intrusion tolerance, a few rounds suffice after consensus on κ-bit inputs. No need for 2 consensus instances as in [14] by Nayak, Ren, Shi, Vaidya and Xiang.

Resilience	Setup	Complexity Overhead
$2t_s + t_a < n$	Trusted	$\mathcal{O}(n^2\kappa + \ell n)$
$2t_s + t_a < n$	None	$\mathcal{O}(n^2\kappa \mathrm{log} n + \ell n)$
$2t_s + t_a \leq (1-\delta)n$	None	$\mathcal{O}(n^2\kappa + \ell n)$

Complexity Summary – Extended

We can reduce the $\mathcal{O}(\ell n^2)$ term to $\mathcal{O}(\ell n)$ with extension protocols.

Thanks to intrusion tolerance, a few rounds suffice after consensus on κ-bit inputs.
No need for 2 consensus instances as in [14] by Nayak, Ren, Shi, Vaidya and Xiang.

Resilience	Setup	Complexity Overhead
$2t_s + t_a < n$	Trusted	$\mathcal{O}(n^2\kappa + \ell n)$
$2t_s + t_a < n$	None	$\mathcal{O}(n^2\kappa \log n + \ell n)$
$2t_s + t_a \leq (1-\delta)n$	None	$\mathcal{O}(n^2\kappa + \ell n)$

When $2t_s + t_a \leq (1-\delta)n$ and the network is synchronous, bulletin-PKI suffices for the complexity $\mathcal{O}(n^2\kappa + \ell n)$ thanks to $(t_s, \delta n)$-intrusion tolerance.

With trusted setup, one can let $\mathrm{CC}_{\mathrm{SBA}} = \mathrm{CC}_{\mathrm{ABA}} = \mathcal{O}(n^2\kappa)$ to achieve network-agnostic consensus with $\mathcal{O}(n^2\kappa + \ell n)$ bits of communication.

References i

[1] Bacho, R., Collins, D., Liu-Zhang, C.D., Loss, J.: Network-agnostic security comes (almost) for free in DKG and MPC. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, Part I. Lecture Notes in Computer Science, vol. 14081, pp. 71–106. Springer, Heidelberg (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_3

[2] Blum, E., Katz, J., Loss, J.: Synchronous consensus with optimal asynchronous fallback guarantees. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019: 17th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 11891, pp. 131–150. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-36030-6_6

[3] Bracha, G.: Asynchronous byzantine agreement protocols. Information and Computation **75**(2), 130–143 (1987). https://doi.org/https://doi.org/10.1016/0890-5401(87)90054-X

[4] Cachin, C., Kursawe, K., Shoup, V.: Random oracles in Constantinople: Practical asynchronous byzantine agreement using cryptography. Journal of Cryptology **18**(3), 219–246 (Jul 2005). https://doi.org/10.1007/s00145-005-0318-0

[5] Clipart Library: Daily envelopes cliparts #3025804: bird with letter clipart, https://clipart-library.com/clipart/n967385.htm

[6] Das, S., Duan, S., Liu, S., Momose, A., Ren, L., Shoup, V.: Asynchronous consensus without trusted setup or public-key cryptography. Cryptology ePrint Archive, Paper 2024/677 (2024), https://eprint.iacr.org/2024/677

[7] Deligios, G., Hirt, M., Liu Zhang, C.: Round-efficient byzantine agreement and multi-party computation with asynchronous fallback. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography — TCC 2021. LNCS, vol. 13042, pp. 623–653. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-90459-3_21

[8] Dolev, D., Strong, H.R.: Authenticated algorithms for byzantine agreement. SIAM Journal on Computing **12**(4), 656–666 (1983). https://doi.org/10.1137/0212045

[9] Flaticon: devil free icon, https://www.flaticon.com/free-icon/devil_725040

References iii

[10] Freitas, L., Kuznetsov, P., Tonkikh, A.: Distributed Randomness from Approximate Agreement. In: Scheideler, C. (ed.) 36th International Symposium on Distributed Computing (DISC 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 246, pp. 24:1–24:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://doi.org/10.4230/LIPIcs.DISC.2022.24

[11] Gao, Y., Lu, Y., Lu, Z., Tang, Q., Xu, J., Zhang, Z.: Efficient asynchronous byzantine agreement without private setups. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). pp. 246–257 (2022). https://doi.org/10.1109/ICDCS54860.2022.00032

[12] Ghinea, D., Goyal, V., Liu-Zhang, C.D.: Round-optimal byzantine agreement. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part I. Lecture Notes in Computer Science, vol. 13275, pp. 96–119. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-06944-4_4

[13] Momose, A., Ren, L.: Optimal communication complexity of authenticated byzantine agreement. In: Gilbert, S. (ed.) 35th International Symposium on Distributed Computing (DISC 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 209, pp. 32:1–32:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). https://doi.org/10.4230/LIPIcs.DISC.2021.32

References iv

[14] Nayak, K., Ren, L., Shi, E., Vaidya, N.H., Xiang, Z.: Improved extension protocols for byzantine broadcast and agreement. In: Attiya, H. (ed.) 34th International Symposium on Distributed Computing (DISC 2020). Leibniz International Proceedings in Informatics (LIPIcs), vol. 179, pp. 28:1–28:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2020). https://doi.org/10.4230/LIPIcs.DISC.2020.28

[15] Toueg, S.: Randomized byzantine agreements. In: Probert, R.L., Lynch, N.A., Santoro, N. (eds.) 3rd ACM Symposium Annual on Principles of Distributed Computing. pp. 163–178. Association for Computing Machinery (Aug 1984). https://doi.org/10.1145/800222.806744