PRACTICAL PROVABLY SECURE FLOODING FOR BLOCKCHAINS

Chen-Da Liu-Zhang, *NTT Research*
Christian Matt, *Concordium*
Ueli Maurer, *ETH Zurich*
Guilherme Rito, *ETH Zurich*
*Søren Eller Thomsen, Aarhus University*
BLOCKCHAINS

Blockchain
BLOCKCHAINS

Blockchain
BLOCKCHAINS

Blockchain
BLOCKCHAINS

Blockchain

A

B
BLOCKCHAINS

Blockchain
BLOCKCHAINS

Blockchain

A
B
C
BLOCKCHAINS

Blockchain

A
B
C

😈

≥
BLOCKCHAINS

Blockchain

A

B

C

Δ-Flood
BLOCKCHAINS

Blockchain

△-Flood
FLOODING FOR BLOCKCHAINS

\[ \Delta \text{-Flood} \]

\[ P_1, P_2, P_3, \ldots, P_n \]
FLOODING FOR BLOCKCHAINS

- Input messages must be delivered within $\Delta$ time.
FLOODING FOR BLOCKCHAINS

- Input messages must be delivered within $\Delta$ time.
FLOODING FOR BLOCKCHAINS

- Input messages must be delivered within $\Delta$ time.
- Assumed to prove security of blockchains [GKL15,PS17,DGKR18,PS18,CM19,DMM+20].
FLOODING IN PRACTICE
FLOODING IN PRACTICE
FLOODING IN PRACTICE
FLOODING IN PRACTICE
FLOODING IN PRACTICE
FLOODING IN PRACTICE
FLOODING IN PRACTICE
BLOCKCHAINS

Blockchain

Δ-Flood

A
B
C
BLOCKCHAINS

Blockchain

A → B → C

\gamma \cdot \# \geq \#
**BLOCKCHAINS**

Blockchain

![Diagram](image)

**Wanted!**

\[ \gamma \cdot \#\text{😊} \geq \#\text{😈} \]
Q: Can efficient flooding be realized assuming a constant fraction of honest weight?
Q: Can efficient flooding be realized assuming a constant fraction of honest weight?

A: YES!
CONTRIBUTIONS

Practical Provably Secure Flooding for Blockchains
Chen-Da Liu-Zhang\textsuperscript{1,}, Christian Matt\textsuperscript{2,}, Ueli Maurer\textsuperscript{3,},
Guillermo Rito\textsuperscript{1,}, and Søren Eiler Thomsen\textsuperscript{4,}
\textsuperscript{1}NTT Research, USA
chen-da.liuzhang@ntt-research.com
\textsuperscript{2}Concordium, Zurich, Switzerland
cm@concordium.com
\textsuperscript{3}Department of Computer Science, ETH Zurich, Switzerland
{maurer, gteixeira}@inf.ethz.ch
\textsuperscript{4}Concordium Blockchain Research Center, Aarhus University, Denmark
sethomsen@cs.au.dk
September 28, 2022

Abstract
In recent years, permissionless blockchains have received a lot of attention both from industry and academia, where substantial effort has been spent to develop consensus protocols that are secure under the assumption that less than half (or a third) of a given resource (e.g., stake or computing power) is controlled by corrupted parties. The security proofs of these consensus protocols usually assume the availability of a network functionality guaranteeing that a block sent by an honest party is received by all honest parties within some bounded time. To obtain an overall protocol that is secure under the same corruption assumption, it is therefore necessary to combine the consensus protocol with a network protocol that achieves this property under that assumption. In practice, however, the underlying network is typically implemented by flooding protocols that are not proven to be secure in the setting where a fraction of the considered total weight can be corrupted. This has led to many so-called collision attacks on existing protocols and take action against specific attacks.

To close this apparent gap, we present the first practical flooding protocol that provably delivers messages to all honest parties after a logarithmic number of steps. We prove security in the setting where all parties are publicly assigned a positive weight and the adversary can corrupt parties accumulating up to a constant fraction of the total weight. This can directly be used in the proof-of-stake setting, but is not limited to it. To prove the security of our protocol, we combine known results about the diameter of Erdős-Rényi graphs with reductions between different types of random graphs. We further show that the efficiency of our protocol is asymptotically optimal.

The practicality of our protocol is supported by extensive simulations for different numbers of parties, weight distributions, and corruption strategies. The simulations confirm our theoretical results and show that messages are delivered quickly regardless of the weight distribution, whereas protocols that are oblivious of the parties’ weights completely fail if the weights are unevenly distributed. Furthermore, the average message complexity per party of our protocol is within a small constant factor of such a protocol.

\textsuperscript{*}Work in part done while the author was at Carnegie Mellon University. Supported in part by the NSF award 1915839, DARPA SEYF program, a gift from Ripple, a DoE-NITL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a CyLab seed funding award.

\textsuperscript{1}Work was in part done while the author was at Purdue University.
1. Weighted Fanout Flooding (WFF):

Abstract

In recent years, permissionless blockchains have received a lot of attention both from industry and academia, where substantial effort has been spent to develop consensus protocols that are secure under the assumption that less than half (or a third) of a given resource (e.g., stake or computing power) is controlled by corrupted parties. The security proofs of these consensus protocols usually assume the availability of a network functionality guaranteeing that a block sent by an honest party is received by all honest parties within some bounded time. To obtain an overall protocol that is secure under the same corruption assumption, it is therefore necessary to combine the consensus protocol with a network protocol that achieves this property under that assumption. In practice, however, the underlying network is typically implemented by flooding protocols that are not proven to be secure in the setting where a fraction of the considered total weight can be corrupted. This has led to many so-called eclipse attacks on existing protocols and talks under them against specific attacks.

To close this apparent gap, we present the first practical flooding protocol that provably delivers sent messages to all honest parties after a logarithmic number of steps. We prove security in the setting where all parties are publicly assigned a positive weight and the adversary can corrupt parties accumulating up to a constant fraction of the total weight. This can directly be used in the proof-of-stake setting, but is not limited to it. To prove the security of our protocol, we combine known results about the diameter of Erdős–Rényi graphs with reductions between different types of random graphs. We further show that the efficiency of our protocol is asymptotically optimal.

The practicality of our protocol is supported by extensive simulations for different numbers of parties, weight distributions, and corruption strategies. The simulations confirm our theoretical results and show that messages are delivered quickly regardless of the weight distribution, whereas protocols that are oblivious of the parties’ weights completely fail if the weights are unevenly distributed. Furthermore, the average message complexity per party of our protocol is within a small constant factor of such a protocol.

---

*Work in part done while the author was at Carnegie Mellon University. Supported in part by the NSF award 1914639, DARPA SIEVE program, a gift from Ripple, a DoD NITL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cybys seed funding award.

†Work was in part done while the author was at Purdue University.
1. Weighted Fanout Flooding (WFF):
   - Secure assuming any constant fraction $\gamma$ of resources being honest.
1. Weighted Fanout Flooding (WFF):

- Secure assuming any constant fraction $\gamma$ of resources being honest.
- Diameter: $O(\log(n))$ for $n$ parties.
CONTRIBUTIONS

1. Weighted Fanout Flooding (WFF):
   - Secure assuming any constant fraction $\gamma$ of resources being honest.
   - Diameter: $O(\log(n))$ for $n$ parties.
   - Message complexity: $O(n \cdot \gamma^{-1} \cdot (\log(n) + \kappa))$.

Practical Provably Secure Flooding for Blockchains

Chen-Da Liu-Zhang\textsuperscript{1,}\textsuperscript{*}, Christian Matt\textsuperscript{2}, Ueli Maurer\textsuperscript{3}, Guillherme Rito\textsuperscript{4}, and Søren Eiler Thomsen\textsuperscript{2,4}

\begin{itemize}
\item \textsuperscript{1}NTT Research, USA
\item \textsuperscript{2}Concordium, Zurich, Switzerland
\item \textsuperscript{3}Department of Computer Science, ETH Zurich, Switzerland
\item \textsuperscript{4}Concordium Blockchain Research Center, Aarhus University, Denmark
\end{itemize}

September 28, 2022

Abstract

In recent years, permissionless blockchains have received a lot of attention both from industry and academia, where substantial effort has been spent to develop consensus protocols that are secure under the assumption that less than half (or a third) of a given resource (e.g., stake or computing power) is controlled by corrupted parties. The security proofs of these consensus protocols usually assume the availability of a network functionality guaranteeing that a block sent by an honest party is received by all honest parties within some bounded time. To obtain an overall protocol that is secure under the same corruption assumption, it is therefore necessary to combine the consensus protocol with a network protocol that achieves this property under that assumption.

In practice, however, the underlying network is typically implemented by flooding protocols that are not proven to be secure in the setting where a fraction of the considered total weight can be corrupted. This has led to many so-called eclipse attacks on existing protocols and fails under them against specific attacks.

To close this apparent gap, we present the first practical flooding protocol that provably delivers sent messages to all honest parties after a logarithmic number of steps. We prove security in the setting where all parties are publicly assigned a positive weight and the adversary can corrupt parties accumulating up to a constant fraction of the total weight. This can directly be used in the proof-of-stake setting, but is not limited to it. To prove the security of our protocol, we combine known results about the diameter of Erdős–Rényi graphs with reductions between different types of random graphs. We further show that the efficiency of our protocol is asymptotically optimal.

The practicality of our protocol is supported by extensive simulations for different numbers of parties, weight distributions, and corruption strategies. The simulations confirm our theoretical results and show that messages are delivered quickly regardless of the weight distribution, whereas protocols that are oblivious of the parties’ weights completely fail if the weights are unevenly distributed. Furthermore, the average message complexity per party of our protocol is within a small constant factor of each such protocol.

\textsuperscript{*}Work in part done while the author was at Carnegie Mellon University. Supported in part by the NSF award 1916999, DARPA TIEVE program, a gift from Ripple, a DoE NITL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cybel seed funding award.

\textsuperscript{1}Work was in part done while the author was at Purdue University.
CONTRIBUTIONS

1. Weighted Fanout Flooding (WFF):
   - Secure assuming any constant fraction $\gamma$ of resources being honest.
   - Diameter: $O\left(\log(n)\right)$ for $n$ parties.
   - Message complexity: $O\left(n \cdot \gamma^{-1} \cdot \left(\log(n) + \kappa\right)\right)$.

2. Extensive simulations of WFF.

---

Practical Provably Secure Flooding for Blockchains

Chen-Da Liu-Zhang\textsuperscript{1}, Christian Matt\textsuperscript{2}, Ueli Maurer\textsuperscript{3},
Guilherme Rito\textsuperscript{1}, and Søren Eilers Thomsen\textsuperscript{4}

\textsuperscript{1}NTT Research, USA
\texttt{chen-da.liuzhang@ntt-research.com}
\textsuperscript{2}Concordium, Zurich, Switzerland
\texttt{cm@concordium.com}
\textsuperscript{3}Department of Computer Science, ETH Zurich, Switzerland
\{maurer, gteixeira\}@inf.ethz.ch
\textsuperscript{4}Concordium Blockchain Research Center, Aarhus University, Denmark
\texttt{sethomsen@cs.au.dk}

September 28, 2022

Abstract

In recent years, permissionless blockchains have received a lot of attention both from
industry and academia, where substantial effort has been spent to develop consensus protocols
that are secure under the assumption that less than half (or a third) of a given resource (e.g.,
slate or computing power) is controlled by corrupted parties. The security proofs of these
consensus protocols usually assume the availability of a network functionality guaranteeing
that a block sent by an honest party is received by all honest parties within some bounded
time. To obtain an overall protocol that is secure under the same corruption assumption,
it is therefore necessary to combine the consensus protocol with a network protocol that
achieves this property under that assumption. In practice, however, the underlying network
is typically implemented by flooding protocols that are not proven to be secure in the setting
where a fraction of the considered total weight can be corrupted. This has led to many
so-called eclipse attacks on existing protocols and take note under them against specific attacks.

To close this apparent gap, we present the first practical flooding protocol that provably
delivers sent messages to all honest parties after a logarithmic number of steps. We prove
security in the setting where all parties are publicly assigned a positive weight and the
adversary can corrupt parties accumulating up to a constant fraction of the total weight.
This can directly be used in the proof-of-stake setting, but is not limited to it. To prove
the security of our protocol, we combine known results about the diameter of Erdős-Rényi
graphs with reductions between different types of random graphs. We further show that the
efficiency of our protocol is asymptotically optimal.

The practicality of our protocol is supported by extensive simulations for different
numbers of parties, weight distributions, and corruption strategies. The simulations confirm
our theoretical results and show that messages are delivered quickly regardless of the weight
distribution, whereas protocols that are oblivious of the parties’ weights completely fail if the
weights are unevenly distributed. Furthermore, the average message complexity per party of
our protocol is within a small constant factor of each such a protocol.

\textsuperscript{1}Work in part done while the author was at Carnegie Mellon University. Supported in part by the NSF award
1914993; DARPA HI-FIVE program; a gift from Ripple; a DoE NITL award; a JP Morgan Faculty Fellowship; a
PNC center for financial services innovation award; and a Cygbus seed funding award.

\textsuperscript{2}Work in part done while the author was at Purdue University.
CONTRIBUTIONS

1. Weighted Fanout Flooding (WFF):
   - Secure assuming any constant fraction $\gamma$ of resources being honest.
   - Diameter: $O\left(\log(n)\right)$ for $n$ parties.
   - Message complexity: $O\left(n \cdot \gamma^{-1} \cdot (\log(n) + \kappa)\right)$.

2. Extensive simulations of WFF.
   - Confirms practicality protocol.
MODEL
Each party $p_i$ has a publicly known weight $w_i > 0$. 

MODEL
Each party $p_i$ has a publicly known weight $w_i > 0$. 
Each party $p_i$ has a publicly known weight $w_i > 0$. 
Each party $p_i$ has a publicly known weight $w_i > 0$.

Assumption: $\exists \gamma \in (0,1], \text{s.t.}$

$\# \bullet \geq \gamma \cdot (\# \bigcirc + \# \bullet)$.
Each party $p_i$ has a publicly known weight $w_i > 0$.

Assumption: $\exists \gamma \in (0,1], \text{ s.t.} \# \bullet \geq \gamma \cdot (\# \bigcirc + \# \bullet)$. 

Implied by the standard PoS assumption.
WARMUP: A SIMPLE INEFFICIENT SOLUTION
WARMUP: A SIMPLE INEFFICIENT SOLUTION

💡 Use existing flooding protocol where parties behave proportionally to their weight.
WARMUP: A SIMPLE INEFFICIENT SOLUTION

Use existing flooding protocol where parties behave proportionally to their weight.

[MNT22]: “Forward to each party with a probability $\rho$” ensures logarithmic diameter.
WARMUP: A SIMPLE INEFFICIENT SOLUTION

💡 Use existing flooding protocol where parties behave proportionally to their weight.

[MNT22]: “Forward to each party with a probability $\rho$” ensures logarithmic diameter.
WARMUP: A SIMPLE INEFFICIENT SOLUTION

Use existing flooding protocol where parties behave proportionally to their weight.

[MNT22]: “Forward to each party with a probability $\rho$” ensures logarithmic diameter.
WARMUP: A SIMPLE INEFFICIENT SOLUTION

- Use existing flooding protocol where parties behave proportionally to their weight.

[MNT22]: “Forward to each party with a probability $\rho$” ensures logarithmic diameter.
WARMUP: A SIMPLE INEFFICIENT SOLUTION

- Use existing flooding protocol where parties behave proportionally to their weight.

[MNT22]: “Forward to each party with a probability $\rho$” ensures logarithmic diameter.
**WARMUP: A SIMPLE INEFFICIENT SOLUTION**

💡 Use existing flooding protocol where parties behave proportionally to their weight.

💡 [MNT22]: “Forward to each party with a probability $\rho$” ensures logarithmic diameter.

$$1 - (1 - \rho)^{w_1 \cdot w_2}$$
WARMUP: A SIMPLE INEFFICIENT SOLUTION

Wanted: Scaling invariance!
A function $E(p)$ that determines how many nodes each party should emulate.
DEVELOPING THE IDEA

💡 A function $E(p)$ that determines how many nodes each party should emulate.
A function $E(p)$ that determines how many nodes each party should emulate.
DEVELOPING THE IDEA

💡 A function $E(p)$ that determines how many nodes each party should emulate.
DEVELOPING THE IDEA

💡 A function $E(p)$ that determines how many nodes each party should emulate.
PROPERTIES OF A GOOD EMULATION FUNCTION
PROPERTIES OF A GOOD EMULATION FUNCTION

- Invariant to scaling of weights.
PROPERTIES OF A GOOD EMULATION FUNCTION

- Invariant to scaling of weights.
- Any party should emulate at least one node.
PROPERTIES OF A GOOD EMULATION FUNCTION

- Invariant to scaling of weights.
- Any party should emulate at least one node.

Message complexity of [MNT22] is linear in $n$ and $\gamma^{-1}$. 
PROPERTIES OF A GOOD EMULATION FUNCTION

- Invariant to scaling of weights.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.

Message complexity of \([\text{MNT22}]\) is linear in \(n\) and \(\gamma^{-1}\).
PROPERTIES OF A GOOD EMULATION FUNCTION

- Invariant to scaling of weights.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.

Message complexity of \([MNT22]\) is linear in \(n\) and \(\gamma^{-1}\).
### CANDIDATES?

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.
CANDIDATES?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E(p) \triangleq w_p$</td>
</tr>
</tbody>
</table>

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.
CANDIDATES?

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.

\[ E(p) \triangleq w_p \]
CANDIDATES?

<table>
<thead>
<tr>
<th></th>
<th>$E(p) \triangleq \alpha_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\triangleq$</td>
</tr>
<tr>
<td></td>
<td>$\alpha_p$</td>
</tr>
</tbody>
</table>

- Invariant to scaling of weight.

- Any party should emulate at least one node.

- Number of emulated nodes should be low.

- Fraction of honestly emulated nodes should be high.
CANDIDATES?

<table>
<thead>
<tr>
<th>Invariant to scaling of weight.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any party should emulate at least one node.</td>
</tr>
<tr>
<td>Number of emulated nodes should be low.</td>
</tr>
<tr>
<td>Fraction of honestly emulated nodes should be high.</td>
</tr>
</tbody>
</table>

$E(p) \triangleq \alpha_p$

Fraction of weight owned by party $p$. 

‣ Invariant to scaling of weight.

‣ Any party should emulate at least one node.

‣ Number of emulated nodes should be low.

‣ Fraction of honestly emulated nodes should be high.
CANDIDATES?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E(p) \triangleq \alpha_p$</td>
<td></td>
</tr>
</tbody>
</table>

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.

Fraction of weight owned by party $p$. 

- $E(p)$ is invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.
<table>
<thead>
<tr>
<th></th>
<th>$E(p) \triangleq \alpha_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fraction of weight owned by party $p$.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>▶ Invariant to scaling of weight.</td>
</tr>
<tr>
<td></td>
<td>▶ Any party should emulate at least one node.</td>
</tr>
<tr>
<td></td>
<td>▶ Number of emulated nodes should be low.</td>
</tr>
<tr>
<td></td>
<td>▶ Fraction of honestly emulated nodes should be high.</td>
</tr>
</tbody>
</table>
CANDIDATES?

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.

\[ E(p) \triangleq \lceil \alpha_p \rceil \]
**CANDIDATES?**

<table>
<thead>
<tr>
<th></th>
<th>( E(p) \triangleq \lceil \alpha_p \rceil )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Invariant to scaling of weight.</td>
</tr>
<tr>
<td></td>
<td>Any party should emulate at least one node.</td>
</tr>
<tr>
<td></td>
<td>Number of emulated nodes should be low.</td>
</tr>
<tr>
<td></td>
<td>Fraction of honestly emulated nodes should be high.</td>
</tr>
</tbody>
</table>
CANDIDATES?

\[ E(p) \triangleq \lceil \alpha_p \rceil \]

- Invariant to scaling of weight. ✓
- Any party should emulate at least one node. ✓
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.
CANDIDATES?

\[ E(p) \triangleq \lceil \alpha_p \rceil \]

- Invariant to scaling of weight. ✓
- Any party should emulate at least one node. ✓
- Number of emulated nodes should be low. ✓
- Fraction of honestly emulated nodes should be high.
**CANDIDATES?**

- Invariant to scaling of weight.  
- Any party should emulate at least one node.  
- Number of emulated nodes should be low.  
- Fraction of honestly emulated nodes should be high.

\[
E(p) \triangleq \left\lceil \alpha_p \right\rceil
\]
CANDIDATES?

<table>
<thead>
<tr>
<th>Condition</th>
<th>$E(p) \triangleq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Invariant to scaling of weight.</td>
<td>✓</td>
</tr>
<tr>
<td>• Any party should emulate at least one node.</td>
<td>✓</td>
</tr>
<tr>
<td>• Number of emulated nodes should be low.</td>
<td>✓</td>
</tr>
<tr>
<td>• Fraction of honestly emulated nodes should be high.</td>
<td>✗</td>
</tr>
</tbody>
</table>
## CANDIDATES?

<table>
<thead>
<tr>
<th></th>
<th>$E(p) \triangleq \lceil \alpha_p \cdot n \rceil$</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Invariant to scaling of weight.</td>
<td></td>
</tr>
<tr>
<td>• Any party should emulate at least one node.</td>
<td></td>
</tr>
<tr>
<td>• Number of emulated nodes should be low.</td>
<td></td>
</tr>
<tr>
<td>• Fraction of honestly emulated nodes should be high.</td>
<td></td>
</tr>
</tbody>
</table>
CANDIDATES?

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.

\[ E(p) ≡ \lceil \alpha_p \cdot n \rceil \]

- Invariant to scaling of weight.
- Any party should emulate at least one node.
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.
**CANDIDATES?**

\[
E(p) \triangleq \left\lceil \alpha_p \cdot n \right\rceil
\]

- Invariant to scaling of weight. ✓
- Any party should emulate at least one node. ✓
- Number of emulated nodes should be low.
- Fraction of honestly emulated nodes should be high.


<table>
<thead>
<tr>
<th>CANDIDATES?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ E(p) \triangleq \lceil \alpha_p \cdot n \rceil ]</td>
</tr>
<tr>
<td>• Invariant to scaling of weight.</td>
</tr>
<tr>
<td>• Any party should emulate at least one node.</td>
</tr>
<tr>
<td>• Number of emulated nodes should be low.</td>
</tr>
<tr>
<td>• Fraction of honestly emulated nodes should be high.</td>
</tr>
</tbody>
</table>
CANDIDATES?

\[ E(p) \triangleq \left\lfloor \alpha_p \cdot n \right\rfloor \]

- Invariant to scaling of weight. ✔
- Any party should emulate at least one node. ✔
- Number of emulated nodes should be low. ✔ (≤ \(2 \cdot n\))
- Fraction of honestly emulated nodes should be high. ✔ (≥ \(2^{-1} \cdot \gamma\))
A FEW ISSUES REMAIN

\[ E(p) \triangleq \lceil \alpha_p \cdot n \rceil \]

\[ 1 - (1 - \rho)^{E(w_1) \cdot E(w_2)} \]
A FEW ISSUES REMAIN

- Selection of neighbors requires \( n \) coinflips.

\[
E(p) \triangleq \lceil \alpha_p \cdot n \rceil
\]
A FEW ISSUES REMAIN

- Selection of neighbors requires \( n \) coinflips.
- Unknown number of neighbors is not very practical.

\[
E(p) \triangleq \left\lfloor \alpha_p \cdot n \right\rfloor
\]

\[
1 - (1 - \rho)^{E(w_1) \cdot E(w_2)}
\]
WEIGHTED FANOUT FLOODING (WFF)
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.
WEIGHTED FANOUT FLOODING (WFF)

1. \( E(p) \triangleq \lceil \alpha_p \cdot n \rceil \)

2. Party \( p \) selects \( K = k \cdot E(p) \) neighbors.
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party $q$ is weighted by $E(q)$. 

Parameter of protocol.
1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party $q$ is weighted by $E(q)$.

Parameter of protocol.
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

Parameter of protocol.

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party $q$ is weighted by $E(q)$.
WEIGHTED FANOUT FLOODING (WFF)

1. \( E(p) \triangleq [\alpha_p \cdot n] \)

Parameter of protocol.

2. Party \( p \) selects \( K = k \cdot E(p) \) neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party \( q \) is weighted by \( E(q) \).

\[
E(P_1) = 2 \\
E(P_2) = 5 \\
E(P_3) = 3 \\
E(P_4) = 4 \\
E(P_5) = 1
\]
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

Parameter of protocol.

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party $q$ is weighted by $E(q)$.
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

Parameter of protocol.

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party $q$ is weighted by $E(q)$.

$E(p_1) = 2$

$E(p_2) = 5$

$E(p_3) = 3$

$E(p_4) = 4$

$E(p_5) = 1$
WEIGHTED FANOUT FLOODING (WFF)

1. $E(p) \triangleq \lceil \alpha_p \cdot n \rceil$

Parameter of protocol.

2. Party $p$ selects $K = k \cdot E(p)$ neighbors.

3. Neighbors are selected by weighted sampling without replacement where each party $q$ is weighted by $E(q)$.
MAIN RESULT
Theorem (informal).

For $k = O\left( (\log(n) + \kappa) \cdot \gamma^{-1} \right)$ and $\Delta = O(\log(n) \cdot \delta)$, $WFF(k)$ is a $\Delta$-Flood protocol.

$\kappa =$ security parameter.  
$\gamma =$ fraction of honest weight.  
$\delta =$ delay on underlying channels.
Theorem (informal).

For $k = O\left((\log(n) + \kappa) \cdot \gamma^{-1}\right)$ and $\Delta = O(\log(n) \cdot \delta)$ WFF($k$) is a $\Delta$-Flood protocol.

- Message complexity: $O(k \cdot n)$.

\( \kappa \) = security parameter.
\( \gamma \) = fraction of honest weight.
\( \delta \) = delay on underlying channels.
Theorem (informal).

For $k = O\left( (\log(n) + \kappa) \cdot \gamma^{-1} \right)$ and $\Delta = O(\log(n) \cdot \delta)$ $WFF(k)$ is a $\Delta$-Flood protocol.

- Message complexity: $O(k \cdot n)$.
- Neighbors of a party $p$: $O\left( k \cdot \lceil \alpha_p \cdot n \rceil \right)$.

$\kappa$ = security parameter.  
$\gamma$ = fraction of honest weight.  
$\delta$ = delay on underlying channels.
PRACTICALITY OF WFF
PRACTICALITY OF WFF

Exp = Exponentially distributed weights.
Rand = Random corruptions.
Heavy = Corrupt heavy nodes first.
Light = Corrupt light nodes first.
WFF VS WOF
WFF VS WOF

“Weight Oblivious Flooding”
WFF VS WOF

“Weight Oblivious Flooding”

Success Rate

Average Messages Sent Per Party

- W*F, Exp(1)
- WOF, Exp(10^3)
- WOF, Exp(10^6)
- WOF, Exp(10^9)
- WFF, Exp(10^3)
- WFF, Exp(10^6)
- WFF, Exp(10^9)
CONCLUSION

- We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.
CONCLUSION

‣ We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.

‣ Many more details and additional results: https://eprint.iacr.org/2022/608.
CONCLUSION

‣ We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.

‣ Many more details and additional results: https://eprint.iacr.org/2022/608.
  ‣ Necessity of increasing neighborhood for heavy parties.
CONCLUSION

‣ We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.

‣ Many more details and additional results: https://eprint.iacr.org/2022/608.
  ‣ Necessity of increasing neighborhood for heavy parties.
  ‣ Necessity of $\log(n)$ neighborhood for fan out flooding.
CONCLUSION

‣ We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.

‣ Many more details and additional results: https://eprint.iacr.org/2022/608.
  › Necessity of increasing neighborhood for heavy parties.
  › Necessity of \( \log(n) \) neighborhood for fan out flooding.
  › Delivery to parties with zero weight.
CONCLUSION

- We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.

- Many more details and additional results: https://eprint.iacr.org/2022/608.
  - Necessity of increasing neighborhood for heavy parties.
  - Necessity of $\log(n)$ neighborhood for fan out flooding.
  - Delivery to parties with zero weight.
  - Additional simulations.
CONCLUSION

‣ We present the first provably secure flooding protocol in the weighted setting and demonstrate its practicality using probabilistic simulations.

‣ Many more details and additional results: https://eprint.iacr.org/2022/608.
 ‣ Necessity of increasing neighborhood for heavy parties.
 ‣ Necessity of \( \log(n) \) neighborhood for fan out flooding.
 ‣ Delivery to parties with zero weight.
 ‣ Additional simulations.

‣ Contact: sethomsen@cs.au.dk.
REFERENCES


SCALABILITY OF WFF

The graph illustrates the success rate of WFF as a function of the average messages sent per party for different numbers of parties. The x-axis represents the average messages sent per party, ranging from 0 to 50, and the y-axis represents the success rate, ranging from 0% to 100%. The graph shows curves for various numbers of parties, including 64, 128, 256, 512, 1024, 2048, 4096, and 8192 parties, each with a different color and marker style.