Mathematical Justification of Attack Impossibility in Secure SDN with Blockchain-Based Trust and Routing Verification

1 Threat Model and Assumptions

We define an adversary A attempting to:

- Compromise an SDN controller to inject malicious flows.
- Forge trust values to be recognized as a legitimate controller.
- Compromise the blockchain to alter flow verification.

Actors:

$$\begin{aligned} C &= \{C_1, C_2, \ldots, C_n\} \quad \text{(Controllers)} \\ S &= \{S_1, S_2, \ldots, S_m\} \quad \text{(Switches)} \\ B &= \{B_1, B_2, \ldots, B_p\} \quad \text{(Blockchain Nodes)} \end{aligned}$$

Assumptions:

- The system maintains t-fault tolerance; an attack succeeds only if A compromises at least t controllers or blockchain nodes.
- The blockchain follows Byzantine Fault Tolerance (BFT) where at most f malicious nodes exist.

1.1 Probability of System Control

To take over the system, the attacker must control:

- At least t_C controllers out of n.
- At least t_B blockchain nodes out of p.

Let:

$$\begin{aligned} P_C(A) &= \text{Probability of compromising an SDN controller} \\ P_B(A) &= \text{Probability of compromising a blockchain node} \\ P_{total}(A) &= \text{Total probability of control} \end{aligned}$$

Using a binomial probability model:

$$P_C(A) = \sum_{i=t_C}^{n} \binom{n}{i} P_C^i (1 - P_C)^{n-i}$$

$$P_B(A) = \sum_{j=t_B}^{p} \binom{p}{j} P_B^j (1 - P_B)^{p-j}$$

$$P_{total}(A) = P_C(A) \cdot P_B(A)$$

If $P_{total}(A) < 10^{-10}$, the attack is mathematically infeasible.

2 Game-Theoretic Attack Analysis

The attacker A chooses a strategy σ_A to maximize $P_{total}(A)$, while the defender D deploys counter-strategies σ_D.

Attacker's payoff:

$$U_A = R - C_A$$

where:

- R = Reward for a successful takeover.
- C_A = Cost of attack.

Defender's payoff:

$$U_D = -U_A$$

At Nash equilibrium:

$$E[U_A] = P_{total}(A)R - C_A < 0$$

If $C_A \gg P_{total}(A)R$, the attack is irrational.

3 Cryptographic Hardness of Blockchain Forgery

For an attacker to rewrite blockchain history, they must solve:

$$H(f_k \| N) < D$$

Probability of mining a block faster than honest nodes:

$$P_{PoW}(A) = \left(\frac{h_A}{h_{total}} \right)^{t_B}$$

where:

- h_A = Attacker's hashing power.

- h_{total} = Total network hashing power.

If $h_A \ll h_{total}$, then:

$$P_{PoW}(A) \approx 0$$

Thus, rewriting history is computationally impossible.

- **Mathematical infeasibility:** If $P_{total}(A) < 10^{-10}$, system takeover is highly improbable.
- **Economic infeasibility:** If $C_A > P_{total}(A)R$, the attack is irrational.
- **Computational infeasibility:** If $h_A \ll h_{total}$, blockchain rewriting is impossible.

This model proves that, under reasonable assumptions, an attacker cannot gain control of the SDN and blockchain-based routing system.

4 Scenario: Coordinated Multi-Vector Attack (Collusion + AI-Augmented Threats)

Imagine an advanced adversary A that combines:

- Insider collusion (rogue admins control some controllers).
- AI-optimized malware that autonomously manipulates network traffic.
- Blockchain forking (trying to create an alternate, malicious chain).

The goal is to control SDN flows and trust values while rewriting blockchain history.

4.1 Probability of Insider Collusion Success

Instead of hacking, the attacker bribes or blackmails insiders. Define:

- I = Number of total administrators.
- I_A = Number of compromised administrators.
- P_I = Probability of bribing or coercing one admin.

The attack succeeds if at least t_I insiders are compromised:

$$P_{\text{collusion}}(A) = \sum_{i=t_I}^{I} \binom{I}{i} P_I^i (1 - P_I)^{I-i} \tag{1}$$

If security policies limit insider risk to 1-2 compromised admins, then for $I = 10$ and $P_I = 0.1$, the probability remains very low.

4.2 AI-Augmented Malware Success Probability

The attacker deploys an AI-driven malware M_A that uses:

- Reinforcement Learning (adapts to network defenses).
- Packet Injection (mimics legit traffic).
- Adaptive Flow Hijacking (re-routes trusted packets).

Let:

- P_M = Probability of bypassing SDN anomaly detection.
- D_M = Defender's ability to detect the malware.

$$P_{\text{malware}}(A) = P_M(1 - D_M) \tag{2}$$

If anomaly detection improves over time ($D_M \to 1$), then $P_{\text{malware}}(A) \to 0$.

4.3 Blockchain Forking Attack Probability

The attacker tries to fork the blockchain to rewrite trust values. Forking requires at least 51% of total computing power. Let:

- h_A = Attacker's mining power.
- h_T = Total blockchain mining power.

$$P_{\text{fork}}(A) = \left(\frac{h_A}{h_T}\right)^{t_B} \tag{3}$$

If the blockchain is widely distributed (e.g., 1000 nodes, attacker controls 5-10), then $P_{\text{fork}}(A) \approx 0$.

4.4 Total Attack Probability

For total system control, all three attacks must succeed simultaneously:

$$P_{\text{total}}(A) = P_{\text{collusion}}(A) \cdot P_{\text{malware}}(A) \cdot P_{\text{fork}}(A) \tag{4}$$

Even with aggressive estimates:

- $P_{\text{collusion}}(A) = 0.05$
- $P_{\text{malware}}(A) = 0.1$
- $P_{\text{fork}}(A) = 0.001$

$$P_{\text{total}}(A) = 0.05 \times 0.1 \times 0.001 = 0.000005 \tag{5}$$

This means the attacker has a 1 in 200,000 chance to fully control the system.

- Collusion is hard due to strict admin policies.
- AI malware needs constant updates, making detection easier.
- Blockchain forks are nearly impossible with strong distribution.

Even with the most advanced attack strategy, the system remains highly secure.

5 Quantum Adversary's Attack Strategy

An attacker A_Q with quantum computing resources aims to:

1. Gain control of the blockchain by controlling 51% of PoW miners or exploiting PoA validators.
2. Modify trust values in the blockchain to override SDN rules.
3. Compromise SDN controllers by sending false instructions.

5.1 Attack Probability Model

The probability of fully controlling the SDN controller depends on:

- Probability of gaining 51% mining power (P_{mine}).
- Probability of corrupting PoA validators (P_{poa}).
- Probability of modifying trust values without detection (P_{trust}).
- Probability of injecting malicious SDN rules (P_{sdn}).

5.2 Step 1: Blockchain Takeover (PoW Attack using Quantum Mining)

If an attacker has quantum mining power $h_A(quantum)$, the probability of controlling 51% of mining at time t is:

$$P_{mine}(A_Q) = \left(\frac{h_A(quantum)}{h_T} \right)^t \tag{6}$$

where:

- h_T is the total mining power of the network.
- $h_A(quantum) = 2 \cdot h_A(classical)$ (due to Grovers algorithm).

If $P_{mine}(A_Q) > 0.51$, the attacker can rewrite blockchain history and control SDN trust values.

5.3 Step 2: PoA Validator Corruption

In Proof-of-Authority (PoA), validators sign transactions. If the attacker can bribe v_A out of V_T validators, then:

$$P_{poa}(A_Q) = \frac{v_A}{V_T} \tag{7}$$

If the majority of validators are corrupted ($P_{poa} > 0.51$), the attacker can approve malicious transactions.

5.4 Step 3: Trust Manipulation and SDN Takeover

The SDN agent trusts the blockchain to verify flows. If the attacker successfully modifies trust values T_{trust}, the probability of undetected manipulation is:

$$P_{trust}(A_Q) = e^{-\lambda \cdot d} \tag{8}$$

where:

- d is the number of blockchain confirmations needed.
- λ is a security constant (higher means stronger blockchain security).

If false trust values are accepted, the SDN controller will install the attacker's flow rules. The probability of this happening is:

$$P_{sdn}(A_Q) = P_{trust}(A_Q) \cdot P_{mine}(A_Q) \cdot P_{poa}(A_Q) \tag{9}$$

5.5 Overall Attack Probability

The final probability of a quantum-assisted blockchain attack fully compromising the SDN controllers is:

$$P_{total}(A_Q) = P_{sdn}(A_Q) = P_{mine}(A_Q) \cdot P_{poa}(A_Q) \cdot e^{-\lambda \cdot d} \tag{10}$$

where:

- Lower d means easier manipulation of blockchain trust.
- Lower λ means weaker blockchain security.
- Higher v_A means more corrupted validators, making PoA takeover easier.
- Quantum miners can reduce PoW difficulty but need 51% mining power.
- PoA corruption is easier than PoW takeover due to fewer validators to bribe.
- Increasing d (blockchain confirmations) and λ (security) reduces attack risk.
- Once the blockchain is compromised, SDN controllers trust it blindly, leading to full network control.