従来のセキュリティ（ゼロトラスト）と新セキュリティ

～無数鍵多重時変成立点理論～との違い

■ ゼロトラストとの本質的な違い

① ゼロトラストの本質

ゼロトラストは：

- 「常に疑う」
- 「毎回検証する」
- 「信頼しない前提で許可する」

つまり、

固定された“正解”は存在するが、それを毎回チェックする

という思想です。

ID・トークン・鍵は存在します。
ただし「継続的に検証する」だけです。

■ 無数鍵多重時変成立点理論の構造の本質

構造は：

1 段階目の鍵 → 通過
副作用チェック
問題なければ 2 段階目生成
さらに副作用チェック
3 段階目生成
… 多層化

さらに重要なのは：

一つの鍵に固定暗号を置かない

毎回ランダム生成される複数暗号化鍵構造

これはゼロトラストとは設計思想が違います。

■ 決定的な違い（構造比較）

項目	ゼロトラスト	無数鍵多重時変成立点理論
正解の存在	固定的に存在	固定しない
セッション	継続前提	成立後即失効
試行回数	増えると危険	増えると不利
鍵	単層 or ローテーション	多層・多鍵・用途別分離
暗号	固定暗号方式	毎回ランダム暗号空間
成立条件	認証成功	条件一致＋副作用無

■ この「第一段階ウイルス通過構造」の意味

ここが極めて重要です。

通常セキュリティは：侵入前に止める・侵入後に検知する

ですが、これは：**成立後に副作用を評価し、次の鍵を決定する**

つまり：「成立＝最終成功」ではない

これは構造的には

段階的成立点制御モデル

L1：通過
↓
副作用解析
↓
L2 鍵生成
↓
挙動解析
↓
L3 鍵生成
↓
用途限定鍵

■ 本理論がゼロトラストでない理由（核心）

ゼロトラストは：「信用しない」

これは：「成立を固定しない」

ゼロトラストは思想レベルの管理強化。
無数鍵多重時変成立点理論は**数学的成立確率の低減構造**。

■ 数理的違い

あなたの資料では：Pdamage = (1−D)(1−S)

さらに：

時間窓 T を短縮
試行回数 N = λT

とすることで

成立確率が急減する構造

ゼロトラストにはこの時間窓概念はありません。

■「固定暗号を置かない」ことの意味

通常：AES 鍵は固定・公開鍵も固定・KMS も固定

無数鍵多重時変成立点理論は：

- 鍵が用途別分離
- 鍵が瞬間生成
- 鍵が成立点に依存
- 再利用不可

これは：**静的鍵モデル → 動的成立空間モデル**

への転換です。

■ ウイルス副作用チェック後に鍵生成する構造

これが最も独自性の高い部分です。

通常は:認証成功 → 全権限発行

無数鍵多重時変成立点理論は:

認証成功
↓
副作用検知
↓
異常がなければ次段階鍵

つまり:「鍵発行＝挙動審査後」

これはゼロトラストの範囲を超えています。

■ 無数鍵多重時変成立点理論の構造を一言で言うと

ゼロトラスト:信用しない前提の管理強化

無数鍵多重時変成立点理論:成立を一瞬の例外にする物理的確率制御

■ 言い換えるなら

ゼロトラスト:常時検証モデル

本理論:成立確率の時間窓圧縮モデル

■ 重要な整理

無数鍵多重時変成立点理論の構造は:「多重鍵」ではなく

「多重成立点 × 多重副作用判定 × 多重暗号空間」です。

ゼロトラストとの差別化は明確になります。

詳細説明書

物理学的成立制御（無数鍵多重時変成立点理論）

― 多層鍵×副作用検査×ランダム暗号空間により「成立そのもの」を瞬間化する ―

---

0. 目的（何を実現するのか）

本方式は、従来の「固定 ID・固定鍵・継続セッション」を前提とした認証／アクセス制御を転換し、
“成立”を一瞬の例外としてのみ許可する構造を採用することで、

- ランサムウェアの**暗号化実行**
- 権限昇格後の**横展開**
- API 悪用による**大量操作**
- 内部不正の**継続濫用**
- AI 自動試行の**累積優位**

を、**成立前／成立直後の段階で遮断**することを狙います

---

1. 従来方式の問題（なぜ破られるのか）

従来モデルには、以下の前提構造が残ります。

- 固定 ID／固定トークン／固定鍵
- セッション継続（ログイン状態の維持）
- 再利用可能（盗まれたら使える）
- 試行回数が攻撃側に有利（AI で自動化できる）

このため「一度突破されると横展開できる」「認証後は暗号化実行が可能」など、**突破後の被害化**を防ぎ切れません

---

2. 本方式の核心（何が違うのか）

2.1 固定の正解を作らない

- “正解（固定 ID・固定鍵）”を守るのではなく、
  成立条件が一致した瞬間だけ、成立空間（有効状態）を生成します
- 成立後は即消滅（再利用できない）

2.2 5 状態モデル（成立空間の状態遷移）

成立空間は以下の 5 状態で遷移します：**無効 → 有効 → 継続 → 失効 ／（異常時）遮断**

---

3. 無数鍵多重時変成立点理論は「多層鍵×副作用検査」構造（詳細）

ここがゼロトラストと最も異なるポイントです。

3.1 多層鍵の基本ロジック

- **第 1 段階鍵：通過（ただし最終許可ではない）**
- **副作用（ウイルス副作用／不正挙動）チェック**
- 問題なければ **第 2 段階鍵を決定・発行**
- 同様に **第 3 段階…** と多重高層化

図解イメージ（概念）

［要求］
　↓
L1：一時成立（入口鍵） ——→（副作用チェック①：挙動/兆候/整合）
　↓ OK
L2：用途限定成立（操作鍵） —→（副作用チェック②：操作の正当性）
　↓ OK
L3：高危険操作成立（特権鍵） →（副作用チェック③：権限濫用/横展開兆候）
　↓ OK
［実行許可］（※許可は最終段の成立後にのみ）

3.2 「固定暗号を置かない」＝ランダム暗号空間

従来は「1 つの鍵＝固定暗号（固定の正解）」になりやすいのに対し、
本方式は **複数暗号化鍵が毎回ランダム**で、段階ごとに鍵の性質（用途・時間・条件）が変わります。
結果として **鍵の“固定的再利用”が成立しにくい**構造になります。

---

4. ゼロトラストとの比較

観点	ゼロトラスト	本方式（成立制御＋多層鍵＋副作用検査）
基本思想	常に検証（Trust しない）	**成立を“瞬間の例外”に限定**
正解の扱い	正解は存在（ID/トークン等）	**固定の正解を作らない**
突破後	権限内で動ける可能性	**副作用検査で次段鍵が出ない**（被害化を止める）
試行回数	多いほど危険	**累積が有利にならない／試すほど不利**
ランサム対策	検知・隔離・復旧中心	**暗号化実行前の成立遮断**
鍵・暗号	ローテーション中心	**多層・多鍵・ランダム暗号空間・用途別分離**

---

5. 攻撃フロー比較（イメージ図）

5.1 従来：突破→横展開→暗号化が通る

侵入 → 認証突破 → 権限取得 → 横展開 → 暗号化実行 → バックアップ破壊

5.2 本方式：暗号化“実行権”が最終段でしか成立しない

侵入 → （L1 成立） → 副作用チェック → （L2 成立） → 副作用チェック → （L3 成立） → 実行
↑異常なら遮断（Blocked）へ

---

6. 数理イメージ（短く・提出向け）

6.1 KPI 定義（資料整合）

- 防御率 D：侵入・成立（突破）を防ぐ確率
- 防衛率 S：突破されても被害化を防ぐ確率
- 被害確率 H：一覧の定義では $H = 1 - S$（73 攻撃一覧に合わせる）

実証評価では総合被害確率として
$P_{damage,1} = (1 - D)(1 - S)$ を使う整理が可能（資料記載）

---

7. 事例（ユースケース別：誰が何をして何が防げるか）

事例 A：ランサムウェア（暗号化・二重脅迫）

従来： 防衛率 55.9% → 被害確率 44.1%
本方式： 防衛率 99.74999999525% → 被害確率 0.25000000475%（設計モデル）
改善倍率：約 176.4 倍

イメージ

- ランサムは「暗号化実行権」が必要
- 本方式は、その実行権が **最終段の成立点**でしか発行されない
- 途中で副作用（横展開兆候／大量操作兆候）が出ると **遮断**へ遷移

事例 B：API 悪用（大量更新・設定変更・データ持ち出し）

想定： API キー漏洩／ログ露出／ソース露出 等

本方式の効き方

- 「API 呼び出し」を L2/L3 の成立対象にする
- **通常 API（閲覧）**は L2
- **破壊的 API（削除・鍵変更・権限変更）**は L3
- L3 は副作用検査が最も厳しく、異常兆候で遮断

事例 C：内部不正（権限濫用・ログ削除・隠密変更）

想定： 権限者が正規資格で不正操作

本方式の効き方

- “ログイン成功”では権限が完成しない
- **操作の種類・頻度・時間帯・対象**で成立条件を分離
- 監査ログ無効化や隠密変更は「副作用」として検知され、
  次段鍵（特権鍵）が出ずに遮断へ

事例 D：災害時・通信断耐性（多経路通報）

多チャネル通報の到達確率モデル：
Pnotify = 1 − (1 − p)^(n(1+r))

（SMS/Push/音声/メール/閉域無線など複数経路で到達率を上げる設計）

---

8. 導入イメージ（システムにどう入れるか）

無数鍵多重時変成立点理論は、既存システムを大改造せず **API 追加ゲート**として上位に被せる考え方が示されています

イメージ（導入方式）

- 既存：アプリ／サーバー／DB／クラウド
- 追加：成立点制御ゲート（API ゲート）
- 追加：KMS/HSM 連携、監査ログ、隔離、バックアップ復旧

要約

- ゼロトラスト：常に検証して“信頼しない”
- 本方式：固定の正解を作らず、“成立”を一瞬だけ許す
- 多層鍵：L1 通過→副作用検査→L2→副作用検査→L3…
- 固定暗号に依存せず、複数鍵がランダム化／用途別分離
- ランサム等の被害化は「暗号化実行権」を成立点で遮断（設計モデル：防衛率 99.7%以上）

具体的なゼロトラストとの比較について

---

1. 結論（まず何が違うか）

ゼロトラストは「常時検証」を強化しますが、基本的に **“固定の正解（ID/トークン/セッション）”を前提**にします。
一方、本方式（無数鍵多重時変成立点理論＋多層鍵＋副作用検査）は **“成立そのものを一瞬に限定して、次段階の鍵発行を副作用（悪性挙動）で止める”** 構造です。
このため、**侵入自体を 0 にする**というより、侵入・通過が起きても **被害化（暗号化・横展開・破壊）を成立させない**方向へ寄せます

---

2. 方式の詳細説明書（マルウェア／ランサム／サーバー攻撃向け）

2.1 従来（ゼロトラスト含む）の弱点が残る理由

従来型の限界として、固定 ID／固定鍵／セッション継続／再利用可能が残り、
「一度突破されると横展開可能」「認証突破後は暗号化実行可能」等の構造的問題が残る、と整理されています。

ゼロトラストは“検証点”を増やしますが、**「検証に通ればその時点で権限が成立する」**設計が多く、攻撃者が一度その条件を満たすと、次の行為（横展開・暗号化・破壊）へ進める余地が残ります。

2.2 本方式の核心：成立空間（5 状態）＋多層鍵＋副作用検査

提出資料の核心はこうです：

- 平常時：成立空間は **存在しない（無効）**
- 条件一致時：**瞬間生成（有効）**
- 正常終了：**即消滅（失効）**
- 異常挙動：**即遮断（遮断状態）**

という **5 状態モデル**

ここに、無数鍵多重時変成立点理論のポイント（ゼロトラストとの差）が入ります

第 1 段階鍵は“通過”し得る
ただし、その後に **ウイルス副作用（悪性挙動）をチェック**
問題なければ **第 2 段階鍵**、さらに…と **多重高層化**
しかも **「1 つの固定暗号鍵」ではなく、複数暗号化鍵がランダムに変化**

この「通過→副作用検査→次段鍵発行」の連鎖が、**マルウェア／ランサム／サーバー侵害**の“被害化の瞬間”を折りにいく設計になります。

2.3 重要：守る対象を「ログイン」ではなく「実行権」に置く

本方式は、ランサム対策の意義として

- **暗号化処理に必要な実行権を成立点制御で遮断可能**
- **同一鍵の再利用不可**
- **成立後即消滅でログ改ざん困難**
- **AI 累積試行無効**
- **横展開無効化**

を挙げています。

つまり、狙いは「ログインを絶対に突破させない」よりも、

暗号化・大量更新・特権操作を成立させる“実行権”を、段階鍵の最終段でしか出さない
途中で副作用が出たら次段鍵を出さず遮断する

です。

---

3. イメージ図(文章で再現:マルウェア／ランサム／サーバー)

3.1 従来(ゼロトラスト含む)の典型フロー

侵入 → 認証突破（or 正規資格の悪用） → 権限成立 → 横展開 → 暗号化/破壊/窃取
↑
ここを超えると “実行” が通りやすい

3.2 本方式(成立点×多層鍵×副作用検査)

侵入（または通過）
↓
L1：入口鍵（通過し得る） ——→ 副作用チェック①（マルウェア兆候/不正挙動）
↓ OK のみ
L2：操作鍵（用途限定） ——→ 副作用チェック②（横展開兆候/大量操作兆候）
↓ OK のみ
L3：特権鍵（暗号化・削除・権限変更など） —→ 副作用チェック③（最厳格）
↓ OK のみ
実行権の発行（暗号化 API / KMS アクセス / 特権操作）
※異常が出た時点で遮断（Blocked）へ

---

4. 攻撃別：詳細事例＋比較

4.1 マルウェア感染（実行ファイル／マクロ等）

73 攻撃一覧での位置づけ

「マルウェア感染」は端末攻撃カテゴリにあり、
従来：防衛率 55.9% → 被害確率 44.1%
本方式モデル：防衛率 99.74999999525% → 被害確率 0.25000000475%
改善倍率：約 176.4 倍（設計モデル）

どう効くか（素直な説明）

- マルウェアは「侵入」自体を 0 にできないケースがある（メール・持込・ゼロデイ等）
- そこで本方式は、**侵入後に“暗号化・横展開・管理者化”に必要な次段鍵（L2/L3）を出さない**ことで被害化を止める
- 副作用チェックでは、例えば
    - 不自然なプロセス連鎖
    - 端末権限の急な昇格
    - 大量ファイル I/O
    - KMS/HSM への異常アクセス
    - 横展開のスキャン挙動
      などを「次段鍵を出さない条件」にし得ます（※実装は PoC で閾値調整が必要）

ゼロトラストで難しい点

ゼロトラストは「誰がアクセスするか」の検証は強い一方、
端末内で動くマルウェアの挙動を前提に“鍵を段階的に出さない”という設計は標準機能ではありません。
（もちろん EDR 連携で近いことはできますが、“鍵発行そのもの”を中心に置くのは別思想です）

4.2 ランサムウェア（暗号化・二重脅迫）

73 攻撃一覧（ランサム行）

従来：防衛率 55.9% → 被害確率 44.1%
本方式：防衛率 99.74999999525% → 被害確率 0.25000000475%
改善倍率：約 176.4 倍

何を“折る”のか

従来では、明確に「ランサムは実行権依存型」と整理されています

つまり攻撃が成立する鍵は：

- 大量暗号化の実行権
- バックアップ破壊の実行権
- 監査ログ無効化の実行権

本方式は、これらを **L3（特権鍵）**に閉じ込め、
L1 や L2 が通っても **副作用があると L3 鍵が出ない**設計にします。

ゼロトラストで不可能だった／難しかった点

- ゼロトラスト自体は「暗号化操作を必ず止める」ことを保証しません
- “正規ユーザの正規端末”が、マルウェアで乗っ取られた場合、検証に通ることがあります
- その後の暗号化実行を止めるには、別途 EDR・権限分離・バックアップ不変化などが必要

本方式はこれらを否定せず、**上位で「実行権を瞬間化」する**と整理されています

4.3 サーバー攻撃（権限昇格／横展開／設定ミス悪用／VPN 悪用）

73 攻撃一覧の該当（例）

脆弱性悪用・設定ミス・認可不備・権限昇格・横展開・ログ改ざん・バックアップ破壊等が並び、
本方式側は同様に **防衛率 99.7499…／被害確率 0.25%** の設計モデルが置かれています

どう効くか（現実的な構造）

サーバー侵害の本質は「横展開」と「特権操作」です。
本方式では、

- **横展開＝同一鍵・同一権限の使い回し** を成立させにくい（再利用不可）
- 管理 API／設定変更／バックアップ削除／権限付与 などを **L3 鍵**に閉じ込める
- 異常兆候（急な権限増大、普段と違う対象への連続操作、スキャン挙動）が出ると遮断

という形になります。

ゼロトラストで難しい点

ゼロトラストは「ネットワーク境界の信頼」を捨てるのが強みですが、
侵入後の“横展開の実行権”を、時間窓と多層鍵で物理的に短くするのはゼロトラストの守備範囲外です。

---

5. 比較表（マルウェア／ランサム／サーバーに限定）

観点	ゼロトラスト	本方式（成立点×多層鍵×副作用検査）
侵入	0 にはならない（前提）	同じく 0 にはできない前提（正直）
侵入後の被害化	別製品・別制御に依存しやすい	**次段鍵を出さない**ことで被害化を折る
ランサム暗号化	“必ず止める”はゼロトラスト単体では言いにくい	**暗号化実行権を L3 鍵に閉じ込め、成立点で遮断**
横展開	最小権限で軽減	**同一鍵再利用不可＋用途別鍵分離**で成立しにくい
鍵の性質	既存の ID/トークン中心	**固定暗号に依存せず、複数鍵がランダム化**（成立後即失効と組）
試行回数（AI 自動化）	検証回数が増えるだけ	**累積優位が成立しない設計**（遮断・条件強化で不利化）

6.「ゼロトラストでは不可能だったのを可能にした」整理

ここは誇張せず、**“ゼロトラストの守備範囲外”**として

6.1 可能にしたこと(守備範囲外を埋める)

1. **実行権(暗号化・破壊・特権操作)を“成立点”で瞬間化する**
   → ゼロトラストは常時検証だが、実行権を時間イベントとして扱う設計は標準ではない
2. **通過を許しても、副作用があれば次段鍵を出さない(多層鍵×副作用検査)**
   → “認証成功=完了”にしない
3. **固定暗号/固定鍵の前提を薄め、複数鍵がランダムに変化する暗号空間**
   → 盗まれた鍵の再利用価値を下げる(成立後即失効と整合)
4. **試行回数が攻撃側の優位にならない(遮断+条件強化)**
   → 数理モデルでも「攻撃が努力するほど不利」方向を記述

6.2 正直な注意点

- 本方式も「実装・運用の設計」が必要です(副作用の判定条件、閾値、誤検知、業務影響の調整)
- “侵入ゼロ”を保証するのではなく、**被害化(暗号化・破壊・横展開)を成立させない**方向が主戦場です
- 既存の WAF/EDR/SIEM/ゼロトラストを否定せず、**上位構造として統合**する立場が資料で明示されています

7.「短い要約」

本提案は、固定 ID・固定鍵・継続セッションに依存する従来構造を転換し、成立を一瞬の例外としてのみ許可する「成立点制御」を導入する。多層鍵(L1→L2→L3)と副作用検査により、侵入・通過が起きても暗号化実行や横展開に必要な実行権を最終段でしか発行せず、異常兆候で遮断へ遷移させる。設計モデルではランサムウェア等で防衛率 99.7%以上、被害確率 0.3%以下を示す(73 攻撃一覧の定義に準拠)。