本資料について

- 本資料は下記論文を基にして作成されたものです。
  文書の内容の正確さは保障できないため、正確な
  知識を求める方は原文を参照してください。

  - 著者: 岡山　聖彦　山井　成良　石橋　勇人
    安倍　広多　松浦　敏雄
  - 論文名: 代理ゲートウェイを用いたSOCKSベースの
    階層的VPN構成法
  - 出展: 情報処理学会論文誌　Vol.42 No.12
  - 発表日: 2001年12月

代理ゲートウェイを用いた SOCKSベースの階層的VPN構成法

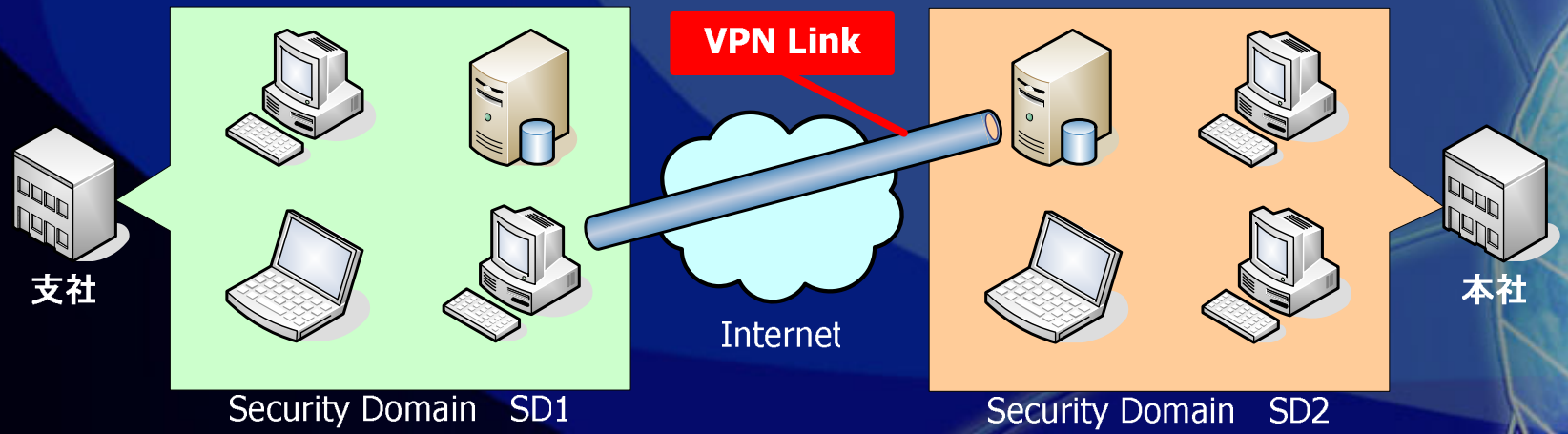
名城大学理工学部情報科学科　渡邊研究室
00J075　　鈴木　秀和

本輪講の題材に使用した論文の紹介

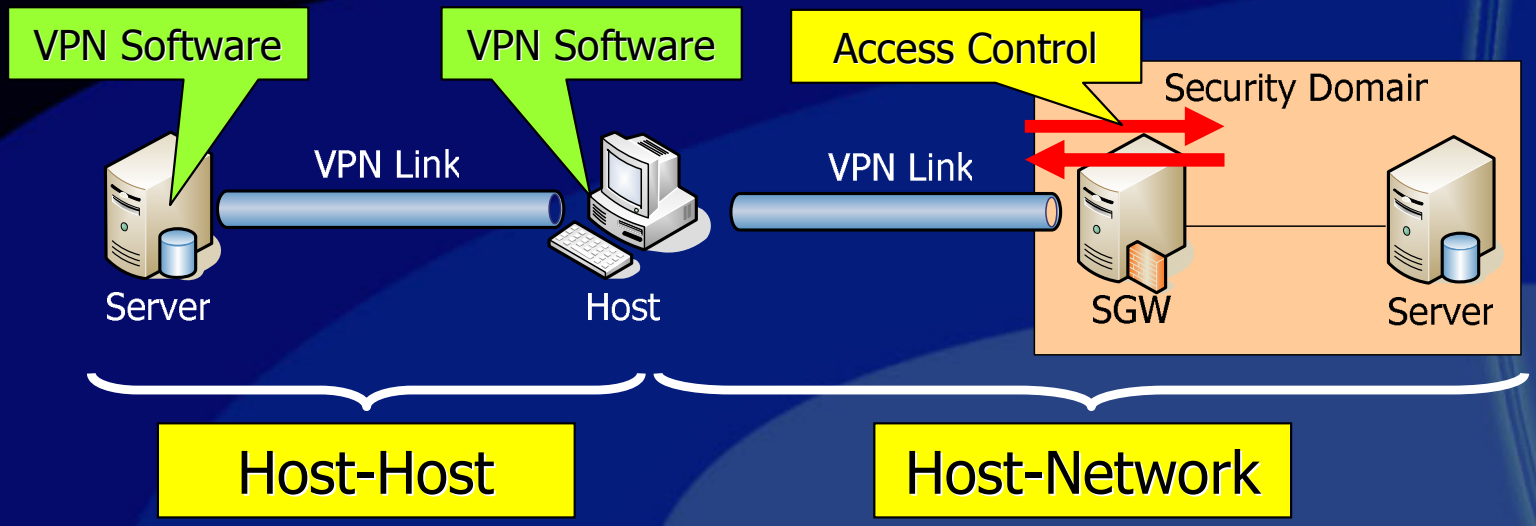
- 代理ゲートウェイを用いたSOCKSベースの階層的VPN構成法
  - 岡山　聖彦　　岡山大学工学部
  - 山井　成良　　岡山大学総合情報処理センター
  - 石橋　勇人　　大阪市立大学学術情報総合センター
  - 安倍　広多　　同上
  - 松浦　敏雄　　同上
- 情報処理学会論文誌　2001.Dec　Vol.42 No.12

はじめに

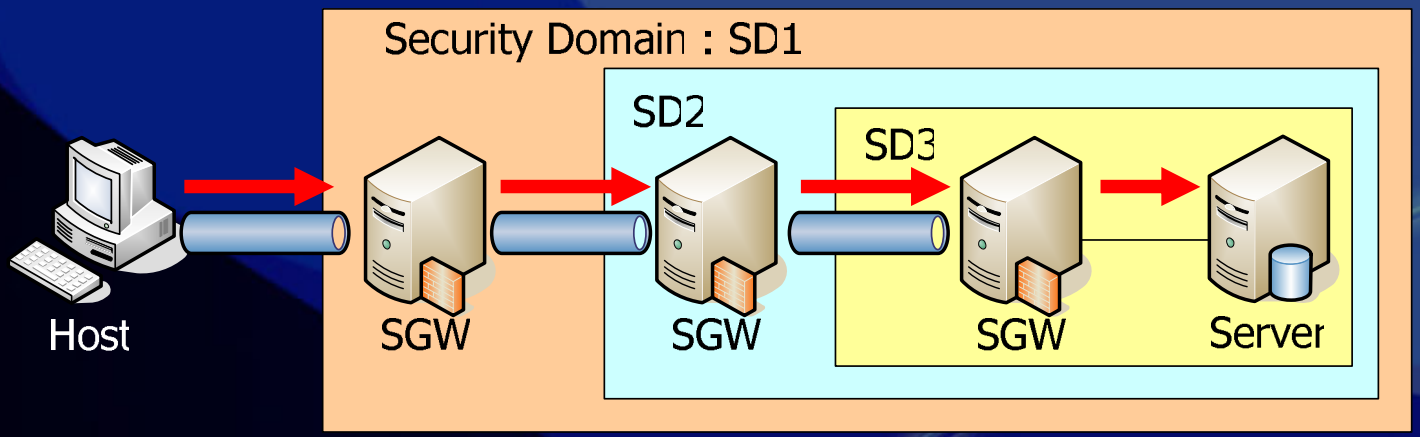
- インターネットの発展にともなってあらゆる情報が流れている
  - » プライバシーに関わる個人情報など秘匿性の高い情報
- 通信の安全性を確保することが重要
  - » 認証および暗号化技術の適用
- VPN（Virtual Private Network）
  - » インターネットを介して遠隔地との間に仮想的リンク（VPN Link）を設けて
    あたかも直接接続されているように見せるための技術



VPNの実現方法



- 大規模な組織では部署ごとにアクセスポリシーが異なる
  - » セキュリティドメインは階層的に構成されるのが自然



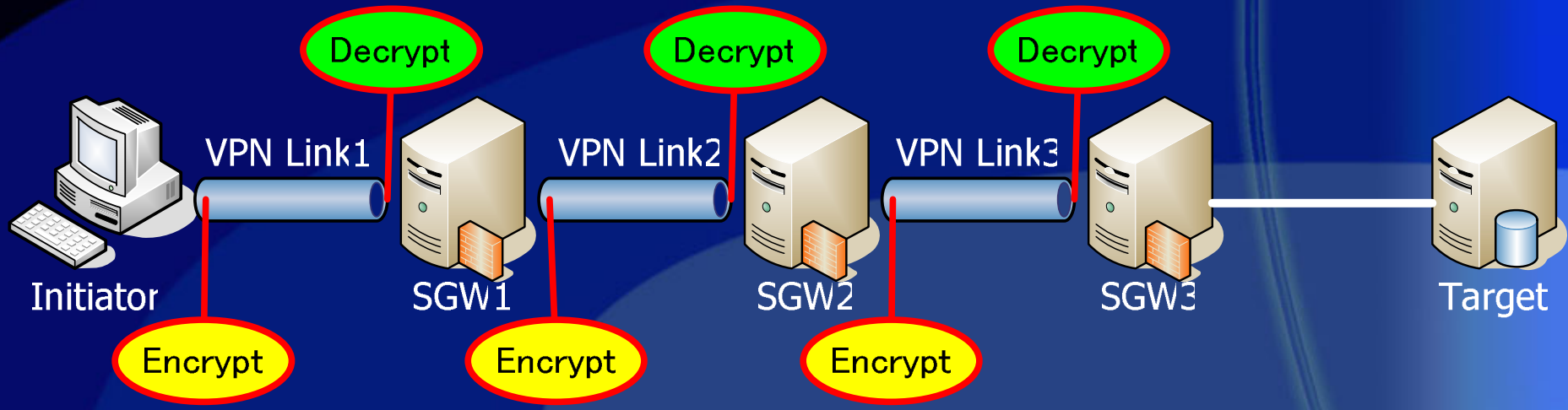
既存のVPNの構成方式

Protocol Layer	Encryption	Non-Encryption
Application	SSH (Remote Login)	
Session	従来法1, 従来法2	
Transport	SSL, TLS, 従来法3, SSH (Port forwarding)	
Network	IPsec, VTun	
Data link	PPTP	L2TP

- SOCKSv5の多段プロキシ機構利用方法　（従来法1）
- SOCKSv5プロトコルの拡張による方法　（従来法2）
- SSL代理サーバをSGWとする方法　（従来法3）

従来法1による構成例と問題点

SOCKSv5の多段プロキシ機構利用方法

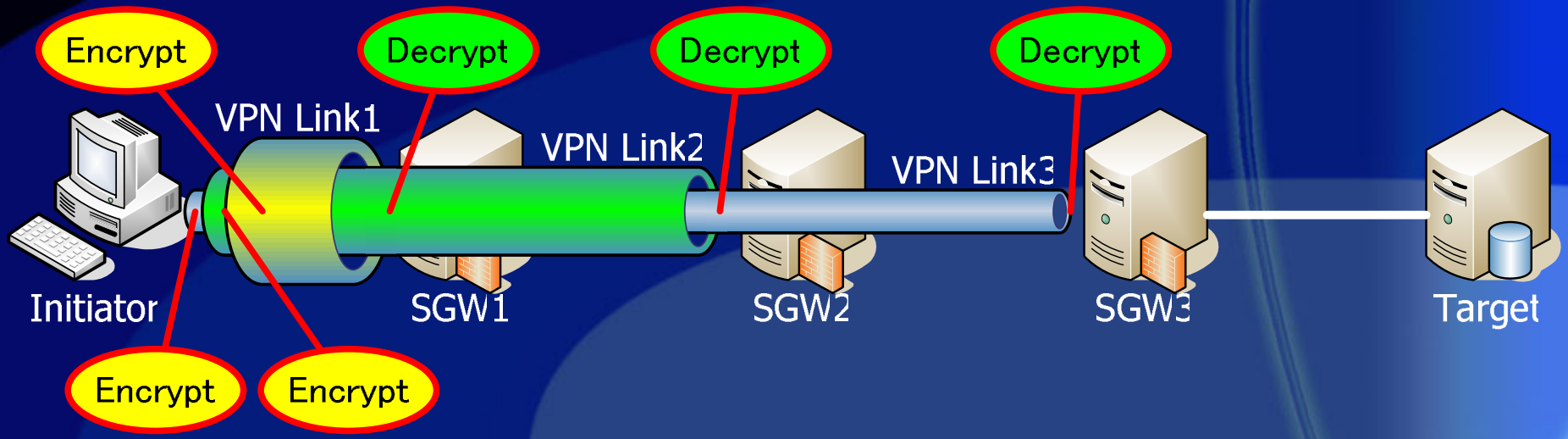


- Initiatorに近い順に隣接するSGWとの間にVPN Linkを確立
- 各VPN Linkごとに独立して認証・暗号化通信が可能
  - » Initiator、各SGWにおいて暗号化・復号が繰り返される

セキュリティドメインの階層数の増加にともない通信効率が低下

従来法2による構成例と問題点

SOCKSv5プロトコルの拡張による方法



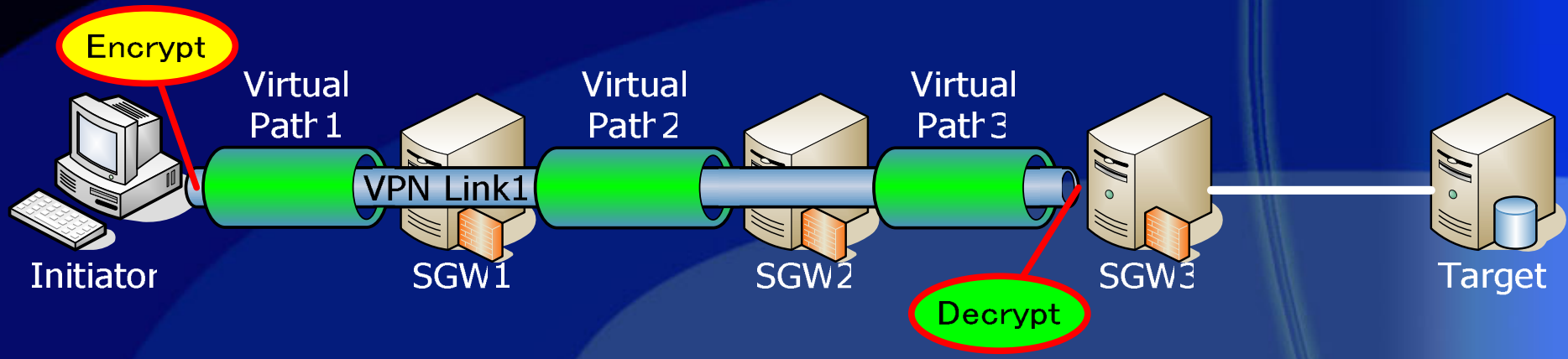
- Initiatorと各SGW間でVPN Linkを確立することを1ホップずつ繰り返す

- 各VPN Linkごとに独立して認証・暗号化通信が可能
  - » Initiator、各SGWにおいて暗号化・復号が繰り返される

セキュリティドメインの階層数の増加にともない通信効率が低下

従来法3による構成例と問題点

SSL代理サーバをSGWとする方法



- 各区間において認証機能のみを持った仮想パスを確立した後、Initiator−終点SGW間のみに1つのVPN Linkを確立
- Initiatorが各SGWと認証を行うためにSSLプロトコルを拡張

Initiatorに対してSSL拡張ライブラリの追加が必須

新しい階層的VPN構成法の前提条件

- 先のような問題を解決するための必須条件

条件1 暗号化のオーバーヘッドが小さいこと

条件2 ユーザに対する利便性を考慮し、イニシエータで既存のVPNソフトウェアがそのまま利用できること

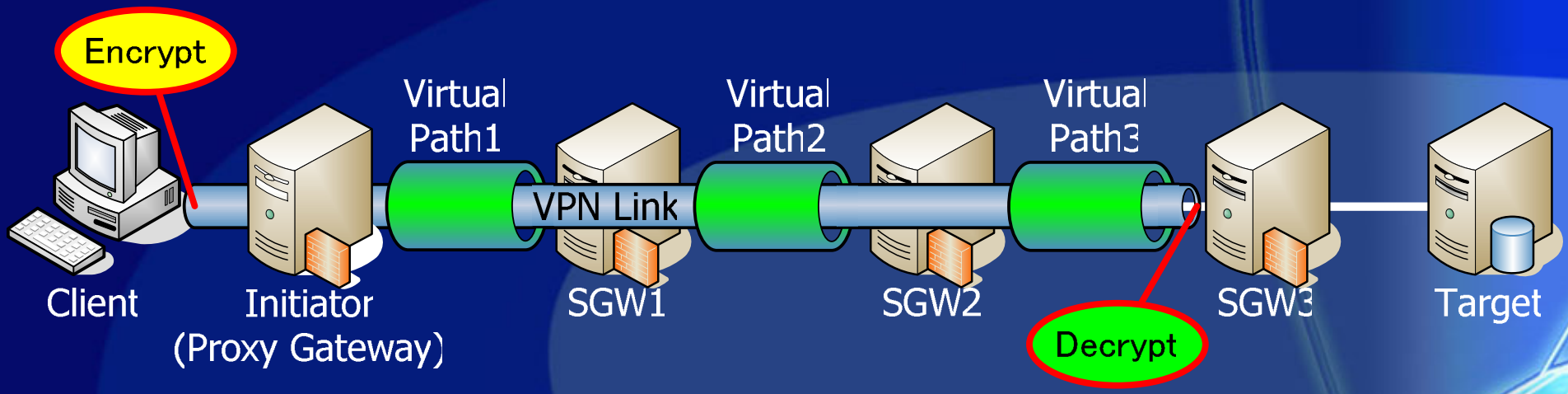
- インターネット環境における管理・運用を考慮した場合の条件

条件3 各SGWごとに独立して認証の有無やユーザ管理などの管理が行えること

条件4 ターゲットが属するセキュリティドメインにおいてプライベートアドレスが利用されていることを考慮し、NATに対応できること

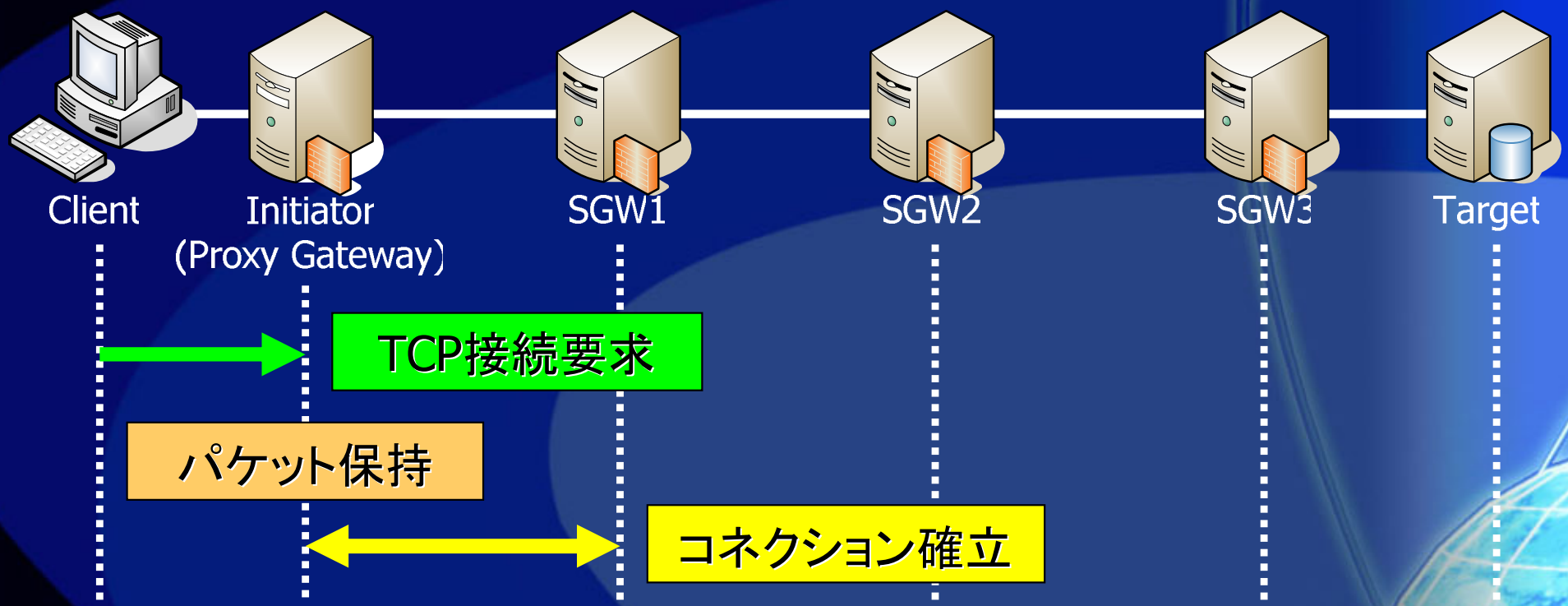
代理ゲートウェイを用いた提案法

- 従来法3のVPN Link確立手順に加え、プロトコル変換を行うための代理ゲートウェイを追加



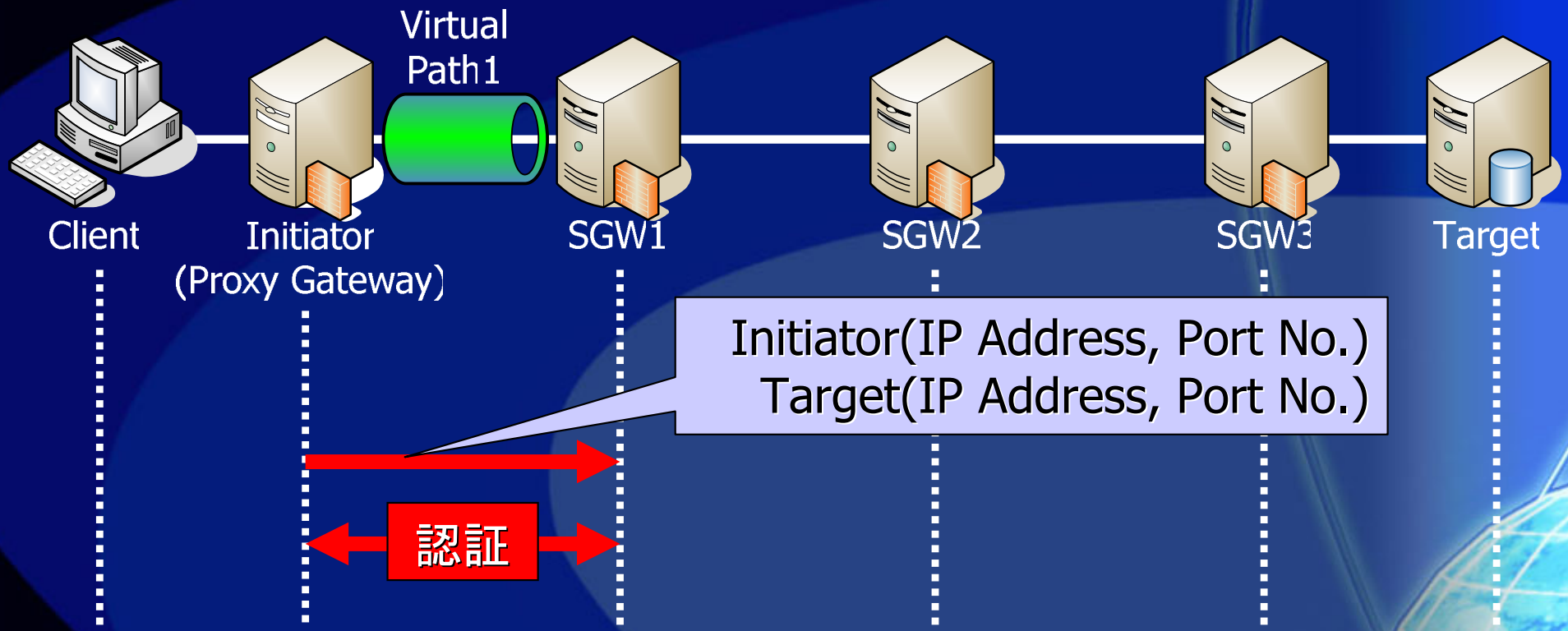
- Clientと始点SGWの間に代理ゲートウェイを設定
- Clientから送信された既存のVPN Link確立要求を代理ゲートウェイが受信
  - » 代理ゲートウェイがInitiatorとなって従来法3と同様の処理により仮想パスを確立
  - » Client–終点SGW間でVPN Linkを確立

提案法によるVPNリンク確立手順



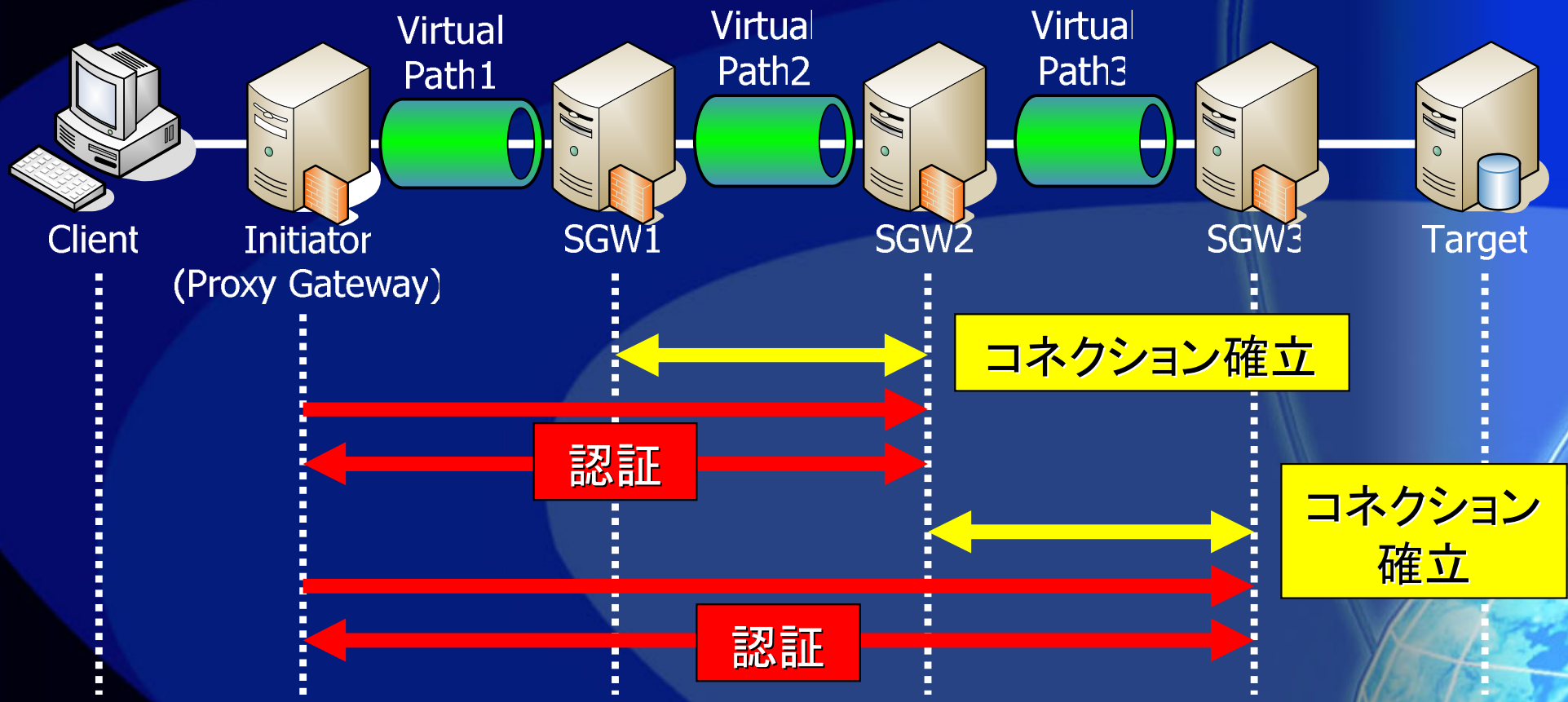
1. Clientは既存のVPN実現方法のプロトコルを用いて、終点SGWとのTCP接続を要求
2. Initiatorはパケット保持後、パケットの宛先アドレスに従って経路表を検索し、セキュリティドメインの最も外側にあるSGWに対してコネクションを確立

提案法によるVPNリンク確立手順



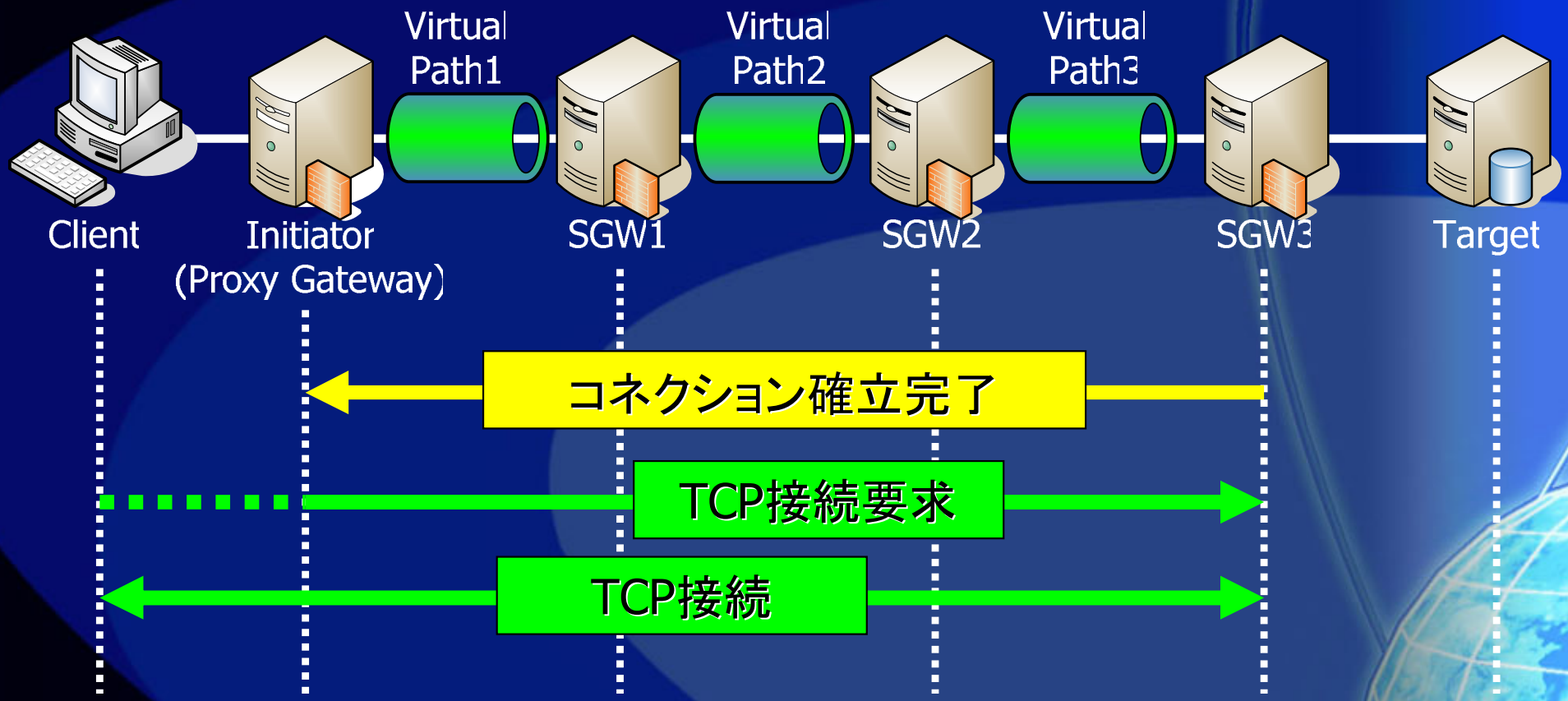
3. Initiatorが必要な情報をSGWに対して送信
4. 受信した情報に基づいて認証方法の決定およびInitiator間で認証を実行
5. 認証成功後、仮想パスができあがりSGWは自己が終端SGWであるかどうかを調査

提案法によるVPNリンク確立手順



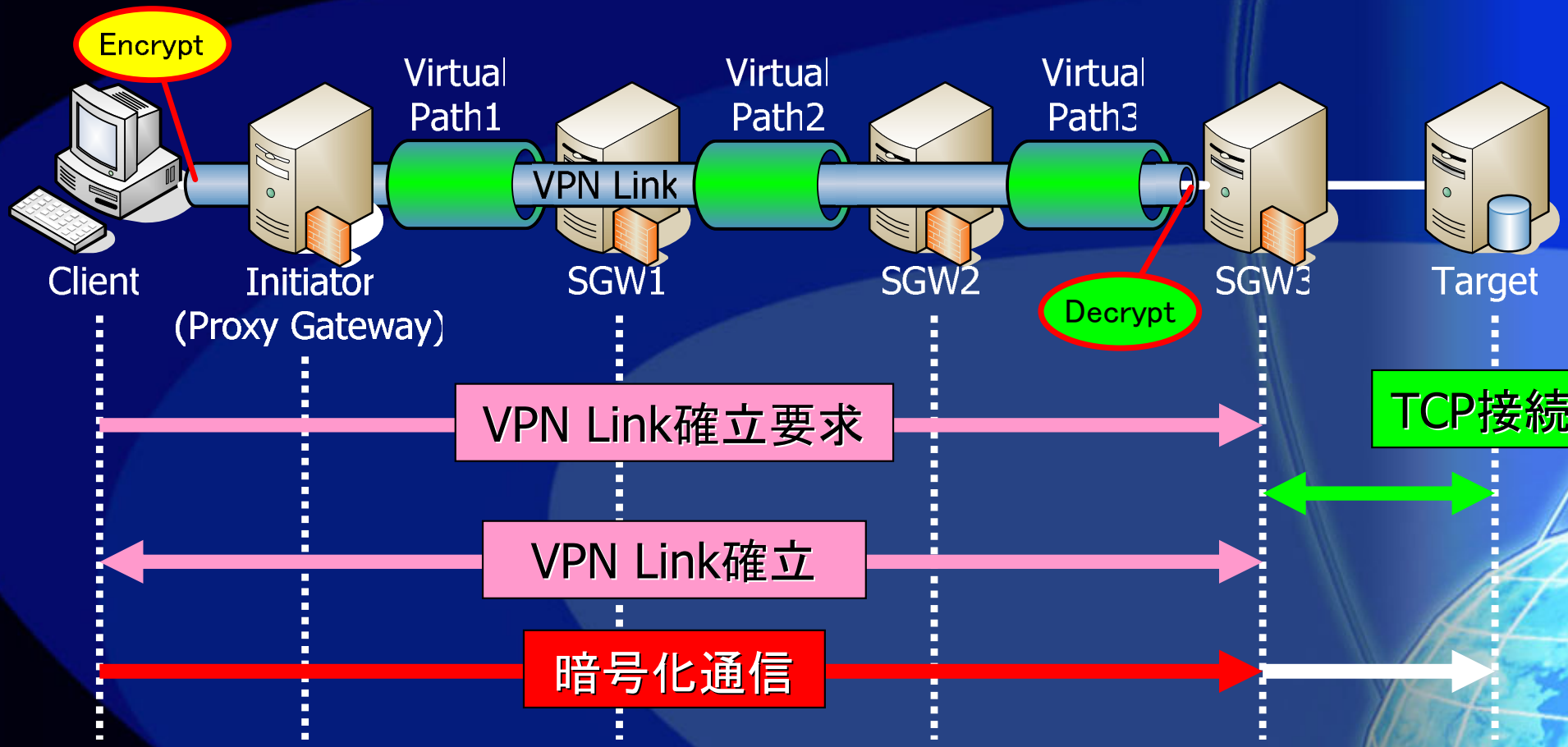
6. 終点SGWでない場合、隣接SGW間にコネクションを確立し、以後パケットを透過的に転送
7. 隣接するSGWをたどりながら終点SGWに到達するまで 3.～6. を繰り返して仮想パスを構築

提案法によるVPNリンク確立手順



8. 終点SGWはコネクション確立完了メッセージを送信
9. Initiatorがメッセージを受信すると保持していたパケットを仮想パスを用いて終点SGWに送信してTCP接続を確保

提案法によるVPNリンク確立手順



10. Clientは既存のVPN実現方法のプロトコルを用いて終点SGWとの間にVPN Linkの確立を要求し成功した場合、終点SGWはClientからのパケットをTargetに転送し、TCP接続を確保
11. Client−終点SGW間で確立されたVPN Linkにおいては、既存のVPN実現方法に基づいて暗号化通信の実行

問題解決の条件を満たしているか？

- VPN Linkを確立するのは常にClient−終点SGW間のみ

条件１　クリア

- 代理ゲートウェイがInitiatorとなって各区間に仮想パスを確立

条件２　クリア

- Initiatorを認証するための情報はSGWのみが知っておけばよい

条件３　クリア

- Targetが属するセキュリティドメインにNATが使用されていても、直接通信するのはInitiatorも含めて隣接する階層のSGW同士

条件４　クリア

実装方法　-SGW-

- 対象：FreeBSD3.2R　4.1R　搭載のAT互換機
- 従来法1のsocksサーバのソースコード（C言語）を変更してSGWおよび代理ゲートウェイを実装

- 従来法1のsocksサーバは認証や暗号化通信の有無の組み合わせをそれぞれmethodとして定義
  - » methodの識別番号を用いてVPN Link確立時に交渉する機能を保持

提案法を示す新たなmethodを定義

（1） 代理ゲートウェイとの認証機能

（2） 認証に必要な次ホップのSGWの情報を代理ゲートウェイに通知する機能

- 交渉によってmethodの選択により従来法1と提案法の両者のSGWとして動作可能

実装方法　-代理ゲートウェイ ・ Client-

- 従来法1のsocksサーバに以下の機能を追加

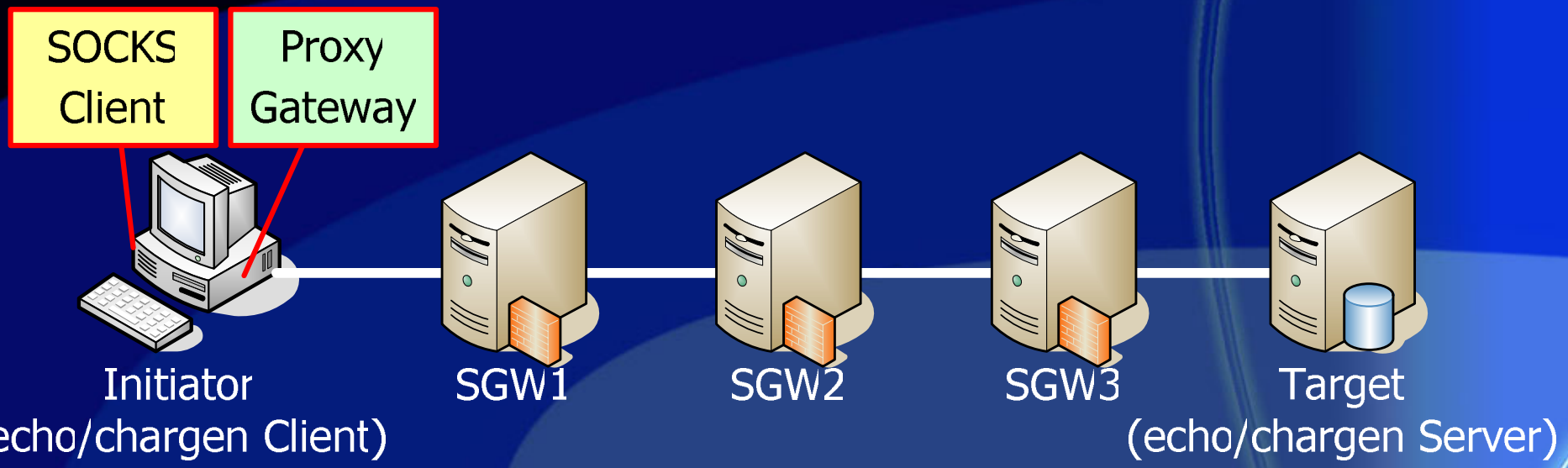
  （1） 提案法を示す新たなmethodの識別番号を用いてSGWと交渉する機能

  （2） 終点SGWに至るまでの各SGWと認証を行うための機能

- Clientについてはsocksクライアントのソースコードの変更無し
- Initiator-各SGW間の認証と、Client-終点SGW間の認証および暗号化通信については、従来法1と同様にKerberos認証および暗号化通信機能を使用

- なぜ実装は従来法1をベースにしているのか？
  - » 提案法のVPN Link確立法は従来法3に基づくが、実装としては従来法1のみが広く公開されている
  - » 従来法1のVPN Linkを仮想パスとして扱うことで実現
  - » アプリケーションに対する組み込みやサービスごとの設定が不要

実験環境

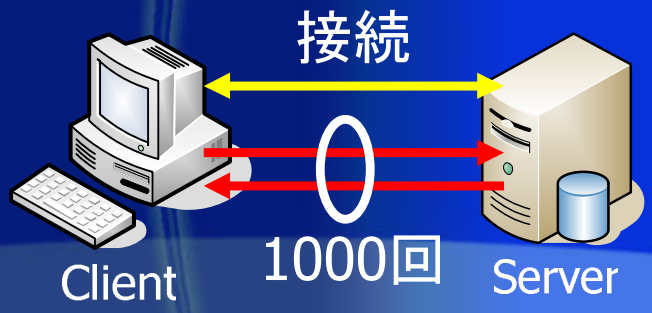


- 提案法と従来法1について実験
- Initiatorのホストにはsocksクライアントと代理ゲートウェイを置き、socksクライアントを代理ゲートウェイあるいは始点SGWのいずれかに接続することにより提案法と従来法1を切り替え
- 各ホストは学内ネットワークを利用（100Mbpsまたは10Mbpsのリンクにより接続）

実験方法

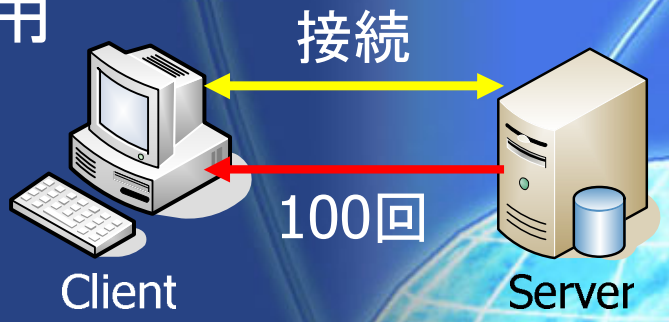
実験1 echo（Port No.7）サービスの利用

- データ量を128，512，1024 [Byte]の3種類
- telnetやrloginなどによる対話的な通信に相当



実験2 chargen（Port No.19）サービスの利用

- データ量を100，500，1000 [Kbyte]の3種類
- ftpなどによるバッチ処理的な通信に相当



- Initiatorでは2つのクライアントを起動して並列実行
  - 起動するクライアントが1つである場合、TCPのslow startの影響により、リンクの帯域に余裕があってもスループットが低下してしまう可能性があり、この影響を軽減するため

実験の計測方法

- VPN Link確立方法(2種類)、データサイズ(各実験とも3種類)の全ての組み合わせに対して実行
  - » 実験1:100回
  - » 実験2:20回
- 各時間の平均値を算出
  - » connect時間:Initiatorの接続開始からVPN Link確立まで
  - » data時間:VPN Link確立後、データ送受信を完了するまで
- いずれの実験においても全てのSGWで認証を行い、かつ全てのVPN Linkで暗号化通信を行う

性能評価

実験1 Data Size [Byte]	従来法1		提案法	
	connect [sec]	data [sec]	connect [sec]	data [sec]
128	1.361	8.631	1.529	5.771
512	1.401	15.210	1.566	9.850
1024	1.409	21.106	1.487	14.995

実験2 Data Size [Kbyte]	従来法1		提案法	
	connect [sec]	data [sec]	connect [sec]	data [sec]
100	1.347	28.158	1.527	26.381
500	1.451	144.204	1.551	131.400
1000	1.383	290.694	1.482	259.829

考察1

- data時間の減少について
  - » 従来法1より提案法の方が小さく、データ量に応じてその差が拡大
    - 暗号化のオーバーヘッドが原因
  - » データ量にかかわらず実験1では従来法1が提案法の約1.5倍、実験2では従来法1が提案法の約1.1倍
    - VPN Link確立方法による差は実験1の方が大きいが、実験1は実験2に比べて1度に送受信するデータ量が小さいため、暗号化のオーバーヘッドによる差がより顕著に現れている
- connect時間の増加について
  - » 従来法1より提案法の方が80～180ミリ秒増加
    - 代理ゲートウェイを追加していることが原因

data時間に比べてconnect時間の差は小さく、実験1や実験2のような手順による通信においては、実用上問題にならないと考えられる

考察2

- 考察1より、提案法は従来法1と比較して暗号化のオーバーヘッドが少なく、より効率的な通信が可能
- 従来法2および従来法3については、原理上、同じ認証方式と暗号化通信方式を用いたと仮定した場合
  - » 従来法2はInitiatorにおいて多重暗号を行うため、従来法1よりも悪化すると考えられる
  - » 従来法3は通信効率は提案法とほぼ同等と考えられるが、Initiatorに対してSSLプロトコルの拡張ライブラリを追加しなければならないため、導入において煩わしい

従来の構成法と比較して提案法の有効性を確認

まとめと今後の課題

- 階層的に構成されたセキュリティドメインにおいて、既存のVPN Link確立方法に対して代理ゲートウェイを導入することで、クライアントに特別なソフトウェアを追加することなく効率的な暗号化通信を実現する方法を提案
- SOCKSv5を拡張することによってSGWと代理ゲートウェイを実装し、実験によりその有効性を確認

- InitiatorおよびSGWが保持する経路表を効率的に管理する方法の検討
  - » 現状では経路表はInitiatorおよび各SGWのそれぞれが設定ファイルとして静的に管理
  - » セキュリティドメインとSGWの対応関係をDNSサーバで管理することにより、SGWの情報をあらかじめ知っておかなくても正しいSGWにアクセスできると考えられる

代理ゲートウェイを用いた SOCKSベースの階層的VPN構成法

終了

SOCKSとは

- 1992年、David Koblas氏の論文で発表
  - » Ying-Da Lee博士（当時 NECシステムズラボラトリー）がSOCKSv4として仕上げる
  - » ネットワークセキュリティの課題に対応するため
- SOCKSの成長
  - » MosaicでSOCKS対応可能
  - » インターネットの普及にともなうプロキシ技術としての開発
  - » SOCKSv4を実装したアプリケーションやサーバが多い
  - » 1995年、SOCKSv5をIETFで標準化の動き（RFC1928）
    - 安全なプロキシ通信に焦点
    - 実装用ソフトウェアは多段プロキシ通信ソフトウェア
    - ファイアウォール技術からVPN技術へ
    - Windowsアプリケーションの動的SOCKS化するDLL（SocksCap）の実現によるWindowsユーザでのSOCKS浸透

SOCKSプロトコルの動作原理