

UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Achieving Secure Routing within Autonomous Systems Independently of Routing Protocols

Permalink

<https://escholarship.org/uc/item/8jf60329>

Journal

Proc. 17th International Conference on Network of the Future (NoF 2026)

Authors

Garcia-Luna-Aceves, J.J.

Moghaddassian, Morteza

Publication Date

2026-09-29

Data Availability

The data associated with this publication are within the manuscript.

Peer reviewed

Achieving Secure Routing within Autonomous Systems Independently of Routing Protocols

J.J. Garcia-Luna-Aceves and Morteza Moghaddassian

Electrical and Computer Engineering Department

University of Toronto

Ontario, Canada (M5S 3G4)

Email: jj.garcialunaaceves@utoronto.ca, m.moghaddassian@utoronto.ca

Abstract—The protocols currently in place for routing within autonomous systems (ASes) of the Internet (e.g., OSPFv2, IS-IS, EIGRP and RIPv2) lack confidentiality and are vulnerable to many cyber attacks. Traditional approaches to adopting security measures in routing protocols rely on either manual configuration of too many symmetric keys or using key-distribution methods that require trust in third parties and digital certificates. This paper describes a new approach to secure routing within ASes that does not make any changes to existing or future routing protocols and completely eliminates the need to use IPsec, digital certificates, manual configuration of link-authentication secrets or passwords, or auxiliary key distribution systems. Trusted routers in the same autonomous system are configured with the same two secret integers, and any two neighbor routers use those secrets to become cryptographically entangled in the use of encryption keys to provide message confidentiality, integrity, and cryptographic authentication. Experimental results show that the new approach to attain secure routing independently of existing or future routing protocols introduces minimum delays for the encryption and decryption of signaling messages among routers.

I. INTRODUCTION

The security threats to routing protocols are well known [10], [15]. Surprisingly, as our review of prior work in Section II describes, the protocols proposed or used today for routing within an autonomous system (AS) of the Internet provide no confidentiality and remain largely unprotected and wide open to attacks.

Security mechanisms have been proposed as part of routing protocols themselves, but they are mostly limited to cryptographic message authentication and integrity checks. The mechanisms themselves rely on either the manual configuration of symmetric keys between a router and each of its neighbor routers, or the use of key-distribution methods that require using other protocol layers and trust on third parties. These approaches have many vulnerabilities, which range from the misconfiguration or lax configuration of trusted routers to the inherent limitations in the handshakes needed to obtain valid public keys or to derive session keys from public keys.

Protocols for routing within ASes provide no confidentiality because they transmit signaling messages in plain text. This enables attacks on a routing infrastructure without having to compromise the routers themselves. As a result, it is much easier for attackers to falsify signaling messages, cause routing tables to hold incorrect information, create service disruptions, and cause performance degradation.

Clearly, a new approach is needed to enable secure routing within ASes of the current Internet and the Internet of the future. We introduce the use of **cryptographic entanglements** as the basis for the design of an efficient and resilient cryptosystem for routing security that is applicable to any current and future protocol for routing within ASes.

Cryptographic entanglements constitute a practical alternative to known key-distribution methods. A *cryptographic entanglement* between two or more routers is a state of correlated processing of information attained by sharing a few secrets that by design differ from the encryption keys used by the routers to protect data. Once routers reach and maintain a cryptographic entanglement, they derive the same sequences of symmetric encryption keys to encrypt and decrypt messages without requiring such messages to be sent reliably and without the routers having to communicate the keys themselves.

Given that we focus on secure routing within a single autonomous system (AS), we assume that system administrators configure in advance all trusted routers with the same few secrets through some secure means.

Section III describes CONCISE (Cryptosystem for Obfuscation, Non-repudiation and Confidentiality with Integer Set Entanglements) as an implementation of cryptographic entanglements that provides secure routing within an AS independently of the routing protocols used in the AS. As such, CONCISE is an attractive approach to secure the routing infrastructure of the Internet of the future.

The trusted routers of an AS using CONCISE are configured with the same two secrets, namely the *Baseline for Entanglement Trust* (BET) and the *jump base*. These two secrets are then used by the routers for the correlated selection and use of encryption keys. A router sending messages to one or multiple neighbor routers become *cryptographically entangled* as soon as the router starts sending encrypted messages to its neighbor router(s) using symmetric encryption keys derived from the shared secrets with which they are configured.

To allow the sending and receiver routers to maintain their correlated selection of encryption keys, the ciphertext sent consists of a header and a payload. The header states the metadata needed for the receivers to use the correct encryption keys without giving any insight on the encryption keys themselves or the structure of the payload, even if messages are lost or arrive out of order. This is done without the need for any

handshake among routers that could divulge any secrets, and without the need to explicitly configure trusted routers with encryption keys.

Any cipher based on symmetric keys can be used to take advantage of the cryptographic entanglements established with CONCISE. However, in this paper, we focus on the use of the Advanced Encryption Standard (AS) [20].

Section IV discusses the small overhead incurred by the maintenance of cryptographic entanglements in CONCISE and presents experimental results showing the performance attained using AES as the cipher in CONCISE. The experiments assume that messages are sent in IPv4 packets with the payload of each packet containing the CONCISE header and the ciphertext payload with routing signaling. The results show that the delay induced by CONCISE is minimum and confirm that CONCISE is a viable approach to secure routing within an AS using *any* current or future routing protocol.

Section V shows that CONCISE provides strong confidentiality, which forces attackers attempting to disrupt routing in an AS protected with CONCISE to focus on subverting the routers themselves, rather than just attacking signaling messages as can be done today. Section VI discusses CONCISE security features, namely: simple configuration of trusted routers, which prevents misconfiguration errors; confidentiality while eliminating the vulnerabilities of current key-distribution approaches; and integrity and authentication enforced by the headers of ciphertexts.

II. SECURITY LIMITATIONS OF INTRA-AS ROUTING PROTOCOLS

The routing protocols operating within ASes of the Internet today primarily provide message integrity and authenticity by relying largely on manual configurations of many symmetric keys and passwords, or the use digital certificates and automated key distribution methods that usually operate on top of the network layer itself and are open to many types of attacks [13], [23], [33], [41]. However, they support no mechanisms to ensure confidentiality, which makes them vulnerable to many passive and active attacks that can be disastrous. The following paragraphs discuss in more detail the limitations of the security measures used by Intra-AS routing protocols.

A. Open Shortest Path First (OSPF) Versions

The Open Shortest Path First (OSPF) protocol [30] is a widely deployed intra-AS routing protocol. OSPFv2 [37] was designed for IPv4 networks and OSPFv3 [16] was designed for IPv6 networks. OSPFv2 supports cryptographic message authentication and integrity checks based on manually configured pre-shared link secrets used by MD5 [37] and HMAC-SHA [11] algorithms to protect against Man-in-the-Middle (MiM) attacks. Some approaches have suggested using digital certificates to distribute link secrets and to sign routing information based on asymmetric cryptography principles [38], [39].

However, OSPFv2 lacks message confidentiality, and OSPF routers communicate in plain text [37]. The lack of confidentiality, the error-prone manual configurations of link secrets,

and the reported use of identical secrets for all OSPF links in a network for ease of configuration have been major sources of insecurity and adversarial attacks on OSPFv2 infrastructures. Examples of such attacks are Remote False Adjacency and Disguised LSA [3], [40].

Compared to OSPFv2, OSPFv3 is less susceptible to adversarial attacks because it supports message confidentiality through IPsec [17], [31]. However, the complexity involved in deploying IPsec often makes message confidentiality inaccessible and costly [43]. IPsec also relies on digital certificates and a Public-Key Infrastructure (PKI) for key distribution, which are prone to known critical security risks [33].

B. Routing Information Protocol (RIP) Versions

RIP is a family of intra-AS routing protocols that are suitable only for small networks, due to their slow convergence time [47]. RIPv1 was designed for IPv4 networks, does not support confidentiality, and has no message authentication method [8], making RIPv1 highly insecure and vulnerable to adversarial attacks such as Route Injection [41].

RIPv2, the current standard version of the RIP protocol for IPv4 networks [8], introduces password authentication as a security enhancement, but it supported in insecure ways [8]. First, RIPv2 embeds plaintext passwords in routing messages which is very insecure, specially in public infrastructures. A password can be easily captured by adversaries. Second, RIPv2 attempts to secure embedded passwords using MD5 or SHA hashing algorithms that are also susceptible to numerous adversarial attacks, such as dictionary [2] and statistical attacks [21].

RIPng was designed for IPv6 networks [30] and provides message authentication and confidentiality through IPsec. However, like OSPFv3, RIPng is susceptible to security threats caused by the lack of IPsec deployments and the reliance on digital certificates in IPsec.

C. Enhanced Interior Gateway Routing Protocol (EIGRP)

EIGRP [5] was developed for routing within large and complex ASes [47]. Compared with OSPF and RIP protocol versions, EIGRP does not support confidentiality unless being tunneled through IPsec. However, it can offer stronger message authentication options using the HMAC-SHA-256 algorithm and the key chains that allow routers to rotate passwords on a timely basis. We note that EIGRP must be used in “Named Mode” to support HMAC-SHA-256. Otherwise, EIGRP “Classic Mode” only supports the insecure MD5 standard, which is open to numerous potential security risks.

D. Intermediate System to Intermediate System (IS-IS) Protocol

IS-IS [24] is widely used by Internet Service Providers (ISPs) for routing within ASes and large data center networks [24]. Unlike other intra-AS routing protocols, IS-IS operates directly over layer two (Data Link), which changes its attack surface by immunizing it against most IP-based adversarial attacks. IS-IS does not support confidentiality but provides

both MD5 and HMAC-SHA-256 message authentication with rotating passwords that protect the routing messages against critical MiM attacks.

E. Proposed Secure Routing Protocols

There have been a number of proposals to secure the signaling of routing protocols that would be applicable for secure routing within an AS (e.g., [9], [13], [28], [45], [48]). However, as it is the case with the routing protocols being used within ASes of the Internet today, these proposals do not provide confidentiality. Furthermore, the proposed methods rely on the use of traditional public-key infrastructures or key distribution methods to support authentication and integrity of routing messages.

III. CONCISE

In contrast to all prior approaches to secure the signaling of routing protocols, CONCISE offers message confidentiality, integrity, authenticity, and ease of use in a way that is *independent* of the routing protocol used in an AS and does not require key distribution, IPsec, or a public-key infrastructure (PKI).

CONCISE establishes and maintains a cryptographic entanglement from a trusted router as a sender and one or more of its neighboring routers as receivers. This is done using common functions applied to the two secrets with which all trusted routers of an AS are configured. Different cryptographic entanglements are established among different sets of neighboring routers to secure the signaling messages of the routing protocol(s) used in an AS.

To ensure that the state of correlated use of encryption keys is maintained even when messages are lost, each ciphertext sent consist of a header and a payload, with the header providing instructions to the receivers on how to select the same encryption keys used by the sending router so that they can decipher the payload even when messages are lost or delivered out of order.

For convenience, we describe the techniques used in CONCISE using router a as the sender in an entanglement and router b as a receiver in an entanglement, and use a and b as their identifiers.

A. Cryptographic Entanglements

The two secrets with which all trusted routers in an AS are configured are the *Baseline for Entanglement Trust* (BET) and the *jump base* (J).

We use $\mathcal{B}$ to denote the configured value of the BET, which is a positive integer of length L_B such that $\mathcal{B} > 2^{L_B-1}$. The value of J is a small integer (e.g., $2 \leq J \leq 100$).

The sender and receivers of an entanglement use a common *anchor function* based on $\mathcal{B}$, J , and the identifier of the sending router a in the entanglement to agree on another integer called the *entanglement anchor*, which is denoted by E . The resulting value of E is such that $E > 2^{L-1}$, where $L > L_B$ is the length of integers used to derive encryption keys in an entanglement.

The length of the integers in $\mathcal{I}$ is much larger than the length of the encryption keys used in the cipher to encrypt keys.

Let $\mathcal{L}_q[n]$ denote the integer corresponding to the q least significant bits (LSBs) of the binary representation of a positive integer n . An example of an anchor function is

$$E = \mathcal{L}_L[(\mathcal{B} + a) \cdot 2^L + J]$$

The entanglement anchor computed by the sender router and receiver routers defines the integer set $\mathcal{I}$ used in the entanglement, which is

$$\mathcal{I} = \{ n \in \mathbb{Z}^+ \mid E \leq n \leq 2^L - 1 \} \quad (1)$$

We denote by k_0 the *basis key* from set $\mathcal{I}$ computed for a message sent from a sender router to the receiver router(s) in the entanglement. The basis key selected for a given message is then used to derive the encryption key or keys needed in the cipher used in CONCISE to encrypt the message, as we discuss subsequently in Section III-D.

B. Relative Anchors and Relative Indices

To cope with packet losses, even if there are many, each message must give the receiver a hint about the basis key k_0 that was used, without revealing its actual value of the keys or divulging E . This hint is based on a relative index in set $\mathcal{I}$. Because set $\mathcal{I}$ may be very large, it is organized into *entanglement groups*, with each group containing 2^R integers and $R \ll L$ to enable the use of relative indices of much smaller length than L . For example, R could be just 32 bits while L could be thousands of bits. The first entanglement group starts with E .

A *relative anchor* is defined to be the integer at the start of an entanglement group, and it is used as the point of reference for the *relative index* stated in a message header in place of a key value. The relative anchor assumed by the sender of an entanglement (router a) in its latest message to router b is denoted by $A(a)$.

The relative index that router a uses in the header of a message to router b instead of a key value is denoted by $R(a)$. Router a maintains the value of the last encryption key used for the last message sent to router b (denoted by ω).

C. Jumping to a New Relative Anchor

The value of E depends on the identifier of the sender, which is the same throughout an AS and may be somehow learned over time by adversaries. To erase this dependency, router a executes a “jump” on the value of the relative anchor used to determine relative indices stated in message headers.

Given that E can be computed at the receiver based on local information, a jump can be done with the first message that router a sends to router b . Once a jump to a relative anchor occur, no router other than a and b can easily decipher messages sent from a to b .

The value of a new relative anchor is computed using a function based on J and a random value selected by router a for which router b is given only a hint that depends on J .

Let $A(a, J)$ denote the relative anchor used by router a just before it proceeds with a jump of its relative anchor. The following is an example of a jump function:

$$A(a) = A(a, J) + 2^R(I \cdot 2^I)J \quad (2)$$

The variable I in Eq. (2) is called the *indicator* of the jump. Router a selects the value of I at random from a small range of values (e.g., $0 \leq I \leq 15$). The indicator is used as an exponent and in combination with J to keep the magnitude of the jump a secret and to reduce the header overhead incurred in giving the receiver the hint about the magnitude of the jump by communicating only the indicator value. If no jump takes place, then $I = 0$.

Router a stores the value of I used to determine the last jump of relative anchors in an indicator record $IR(a)$, and this value is used in message headers.

Figure 1 illustrates the use of a jump to change the relative anchor based on Eq. (2). In the example, router a executes a jump of $(I \cdot 2^I)J$ entanglement groups with its first message to router b . The example also illustrates the fact that the hint provided by router a to router b regarding the value of the first key used for the encrypted message in the payload is based on the agreement that a and b maintain on the relative anchor used as a reference point for the values of $R(a)$ and $IR(a)$ stated in the message header. The meaning of variable F in the figure is described below.

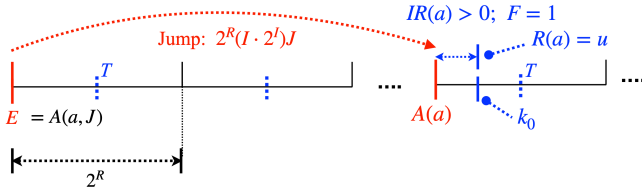


Fig. 1. A jump to a new relative anchor

D. Selecting Encryption Keys for Each New New Message

The length of integers in $\mathcal{I}$ is large enough to accommodate any of the ciphers that can be used in a deployment of CONCISE. The encryption keys used in a message are obtained from a basis key in set $\mathcal{I}$ and the length of the keys is determined by the cipher used in CONCISE.

A sender router a selects a new basis key k_0 from set $\mathcal{I}$ for each new message it sends in an entanglement using a function based on J and a random selection made in such a way that k_0 is larger than the last basis key used.

Let $v \in [1, m]_U$ denote an integer v chosen at random from the interval of equally likely integers $[1, m]$. An example of a function that router a can use to select a basis key k_0 is given in Eq. (3).

$$k_0 = \begin{cases} A(a) + R_K(a) + u & \text{if } IR(a) = 0 \\ A(a) + u & \text{if } IR(a) > 0 \end{cases} \quad (3)$$

with $u \in [1, 1 + J\mathcal{L}_8[\omega]]_U$

A cipher used in CONCISE may require multiple keys to encrypt a single message, and all such keys can be derived

from the basis key k_0 using a common function. In this paper, however, we assume that AES is used as the cipher in CONCISE, because it is widely used today and has been shown to provide strong security.

With AES as the cipher in CONCISE, the payload of a CONCISE message is treated as a single message block and a single encryption key is needed for each message that is to be encrypted. We denote this encryption key by k_1 , which is obtained from k_0 using a key-selection function.

Let L_c denote the length of the encryption keys required by the AES cipher used in CONCISE, a simple example of a key-selection function in this case is the following:

$$k_1 = \mathcal{L}_{L_c}[k_0] \quad (4)$$

E. Correlating Keys and Jumps without Reliable Handshakes

CONCISE does not do any retransmissions and does not use any handshakes for the sender and receivers of an entanglement to agree on encryption keys. If a packet carrying a routing message is lost, the routing protocol is in charge of its retransmission and the new packet is protected with new encryption keys.

Instead of reliable handshakes, CONCISE uses a persistence approach to ensure that both ends of an entanglement agree on a new relative anchor after either the sender executes a jump of relative anchors or the value of k_0 is larger than $A(a) + 2^R$. The approach consists of the receiver using the relative anchor in place before the jump or the needed change of relative anchor based on the indicator value I , which is persistently shared by router a with router b , until the keys used in messages are larger than a *threshold index value* T relative to the new relative anchor. To make the approach highly tolerant to many packets being lost, the value of T is one half the size of an entanglement group, i.e., $T = 2^{R-1}$.

The message header includes a one-bit *key flag* denoted by F to inform the receiver about the transition from the current relative anchor to the next in sequence (i.e., from $A(a)$ to $A(a) + 2^R$). This flag is needed because the indicator I of a jump can only denote large jump values.

Algorithm 1 Steps at Sender a

- 1: $R_K(a) \leftarrow k_0 - A(a)$;
 - 2: **if** $A(a) \leq k_0 < A(a) + T$ **then** $F \leftarrow 1$ **else** $F \leftarrow 0$
 - 3: **if** $k_0 \geq A(a) + T$ **then**
 - 4: $A(a, J) \leftarrow A(a)$; $IR(a) \leftarrow 0$
 - 5: Message $\leftarrow (R(a), IR(a), F)$
-

Algorithm 2 Steps at Receiver b

- 1: **if** $(IR(a) > 0) \wedge (IR(b) \neq IR(a))$ **then**
 - 2: $IR(b) \leftarrow IR(a)$; $A(b) \leftarrow A(b, J) + 2^R(IR(b) \cdot 2^{IR(b)})J$
 - 3: **if** $IR(a) = 0$ **then**
 - 4: $IR(b) \leftarrow IR(a)$;
 - 5: **if** $(F = 1) \wedge (f = 0)$ **then**
 - 6: $f \leftarrow 1$; $A(b) \leftarrow A(b) + 2^R$
 - 7: $k_0 \leftarrow R(a) + A(b)$
-

Steps at the sender: When router a uses Eq. (2) to jump to a new relative anchor, it stores the value used for I in $IR(a)$, and can start a jump only when its current value of $IR(a)$ is zero, i.e., router a does not carry out multiple consecutive jumps. After router a uses Eq. (3) to obtain the value of k_0 for a message to router b in the entanglement, it executes Algorithm 1 to determine the values of the hints included in the message header.

Steps at the receiver: Router b executes Algorithm 2 when it receives a message from router a . The value $A(b)$ is the current relative anchor assumed by router b ; $A(b, J)$ is the stored value of the relative anchor at router b for the last message that b received from a stating $IR(a) = 0$; and f and $IR(b)$ are the stored values at router b of the key flag and the indicator record, respectively.

Figure 2 illustrates the use of the key flag F . The example assumes that $IR(a) = 0$, i.e., the sender did not carry out a jump when computing the value of k_0 for the new message.

In the example shown in Figure 2(a), k_0 is below the threshold T and k_0 is beyond the threshold in the example of Figure 2(b). This does not confuse the receiver, because the values of F used at the sending router a and the local value of f at the receiver router b are taken into account in Algorithm 2 at the receiver router b , and the local value of f is updated to the value of F the first time a difference between F and f is detected (Line 5 in Algorithm 2) and the value of the relative anchor is changed only once as values of the basis keys become larger than the current threshold value. The value of the encryption key k_1 is selected using a simple key-selection function like the one shown in Eq. (4).

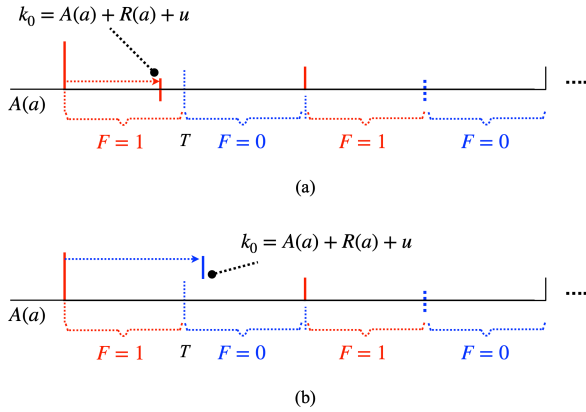


Fig. 2. Use of key flags in the selection of basis keys

F. Reusing the Integers from $\mathcal{I}$ in an Entanglement

When the current relative anchor equals $2^L - 2^R$, the next relative anchor to use is E , which remains unchanged. This is illustrated in Figure 3.

As the figure illustrates, using E as the new relative anchor amounts to a jump of relative anchors. However, the target value (E) is known and is assumed to be the same as incrementing the value of relative anchor by a jump of one entanglement group. Accordingly, the CONCISE messages

from the sender do not have to indicate the jump explicitly. As the example in Figure 3 illustrates, it is an implicit jump of relative anchor back to E . With this approach, the basis keys used to derive encryption keys always have length L .

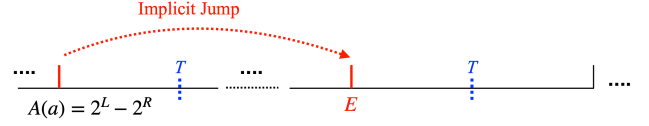


Fig. 3. Reuse of integers in set $\mathcal{I}$ when relative anchor equals $2^L - 2^R$

G. Point-to-Multipoint Entanglements

In any routing protocol a router may communicate with multiple neighbor routers over a broadcast medium (e.g., Ethernet). In this case, a sender router needs to establish a point-to-multipoint entanglement with the neighbor routers it has in the broadcast medium. Establishing point-to-multipoint cryptographic entanglements in CONCISE is remarkably simple. Receiver routers simply listen to broadcast or multicast addresses used in the signaling messages by a sending router. A sender router sends the same message to the broadcast or multicast address and can execute a jump to a new relative anchor to be used with all its neighbor routers without the need for reliable handshakes with neighbor routers.

A sender router that needs to send the same messages to all or a subset of its neighbours simply selects a *multipoint relative anchor* for its multipoint entanglement with multiple neighbor router. It then transmits its broadcast or multicast message to them to the known multipoint address to which the intended receivers listen. Its message reflects the jump to the multipoint relative anchor, and the rest of the steps are the same as with point-to-point entanglements.

IV. CONCISE OVERHEAD

A. CONCISE Message Header

The CONCISE header is embedded in every routing message payload in the network, is just 41-byte long, transmitted unencrypted, and states the metadata needed by the recipient routers to decipher encrypted routing messages. The elements of the CONCISE header are:

Relative Key Index (32 bits): States $R(a)$.

Indicator of Jump (4 bits): States the value of $IR(a)$.

Key Flag (4 bits): States the value of F set according to Algorithm 1.

Message Timestamp (32 bits): Denotes the time when the Message Hash is computed and helps preventing replay attacks by stating the timeliness of the message.

Message Hash (256 bits): It is a SHA-2 hash of the CONCISE header and the plaintext routing message data and is primarily used for message integrity and authentication purposes. Unlike integrity tags used by routing protocols (see section II), this message Hash is tied with the cryptographic entanglement keys that are not predefined and requires trusted routers to first decrypt the message data to be able to reproduce the Hash. Therefore, it is protected against MiM attacks.

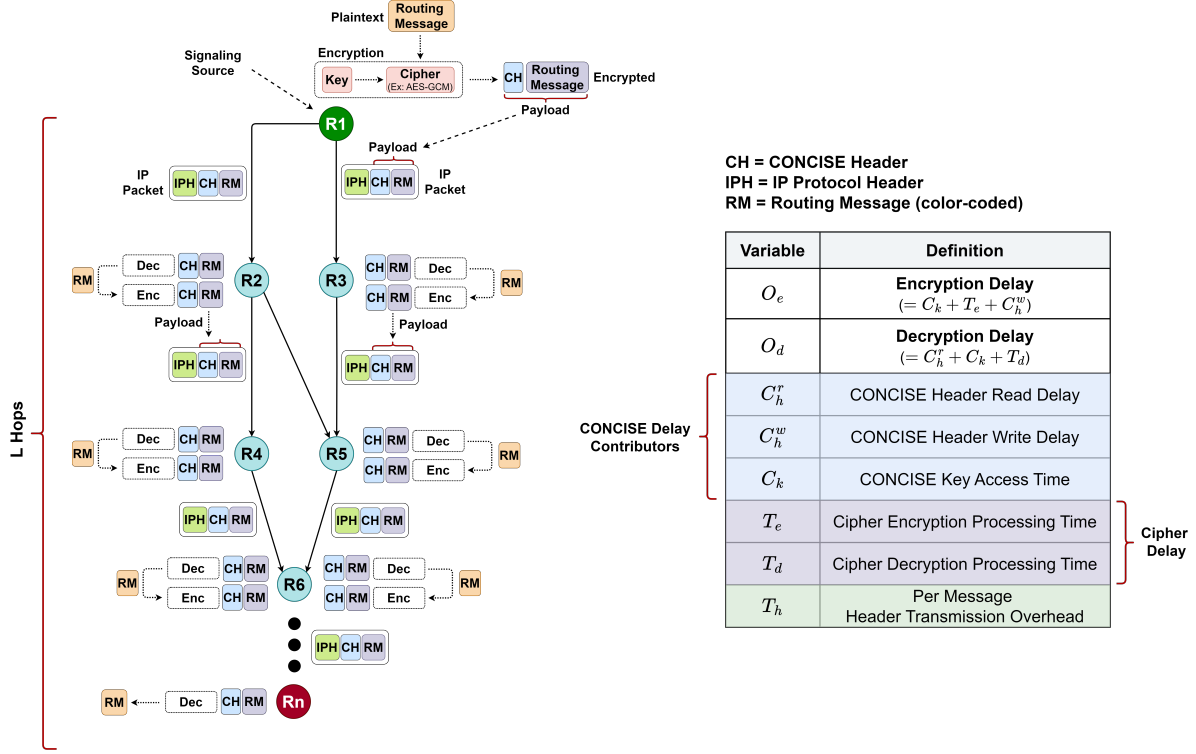


Fig. 4. Hop-by-Hop overhead of adopting CONCISE in a simple IP network

B. Encryption and Decryption Overhead

The delay overhead induced by CONCISE consists of the transmission delay resulting from including the CONCISE header (T_h) in each routing message and the overhead of the message encryption and/or decryption carried out by individual routers. Fig. 4 illustrates the hop-by-hop overhead incurred by using CONCISE in a network.

The components of the encryption delay (O_e) consists of the key access time (C_k), the processing time needed by the cipher (T_e) to encrypt the plaintext, and the time it takes for the router to compile the CONCISE header after the message is encrypted (C_h^w). Similarly, the components of the decryption delay (O_d) are the time needed by the message receiver to read the CONCISE header (C_h^r) plus C_k and the time needed by the cipher to decrypt the ciphertext (T_d).

We assume that the value of C_k is approximately the same in O_e and O_d as once a message receiver reads the CONCISE header, the keys are known to the cipher. C_h^w and C_h^r typically take sub-microseconds in standard computers as the keys are always ready at the source and the header is very small to compile. On the other hand, T_e and T_d may vary across various ciphers, as shown in Fig. 5.

Depending on whether a router is the source of a routing message, a relay router that consumes and forwards the routing messages, or the last router in a forwarding path (e.g., R_n in Fig. 4), the total overhead can be slightly different. For a source router (e.g., R_1 in Fig. 4), the total overhead is calculated as $T_o = O_e + T_h$ which also includes the time it

takes for an encrypted routing message to be delivered to the next router (T_h). As one would expect, the processing of the routing messages at the source does not include a decryption overhead.

A relay router, on the other hand, must not only decrypt a message based on the entanglement keys it has with its previous router in the forwarding path to access a copy of the message data but must also re-encrypt the message based on its entanglement keys with the next router in the forwarding path to send an encrypted copy of the message forward. Therefore, for an intermediary router, the total overhead of adopting CONCISE is calculated as $T_o = O_d + O_e + T_h$. The last router in a forwarding path must only decrypt the message without the need for re-encryption. For this router the total overhead of adopting CONCISE is calculated as $T_o = O_d$. Finally, to calculate the total overhead of CONCISE after certain number of hops in the network, one would take the number of hops as a multiplier. For instance, the total overhead of a routing message sent by R_1 and received by R_5 in Fig. 4 is $T_o = 2 \times (O_e + O_d + T_h)$ assuming that R_5 must also decrypt the message to access its routing data.

As we have explained, the added delay induced by CONCISE in the overall overhead of encrypting and decrypting routing messages is minimal (sub-microsecond). The main delay overhead comes from the cipher. To provide an idea about this delay, we conducted an experiment in which we measured the performance of selected symmetric-key ciphers [6] used commonly in encrypting and decrypting IP packets that carry

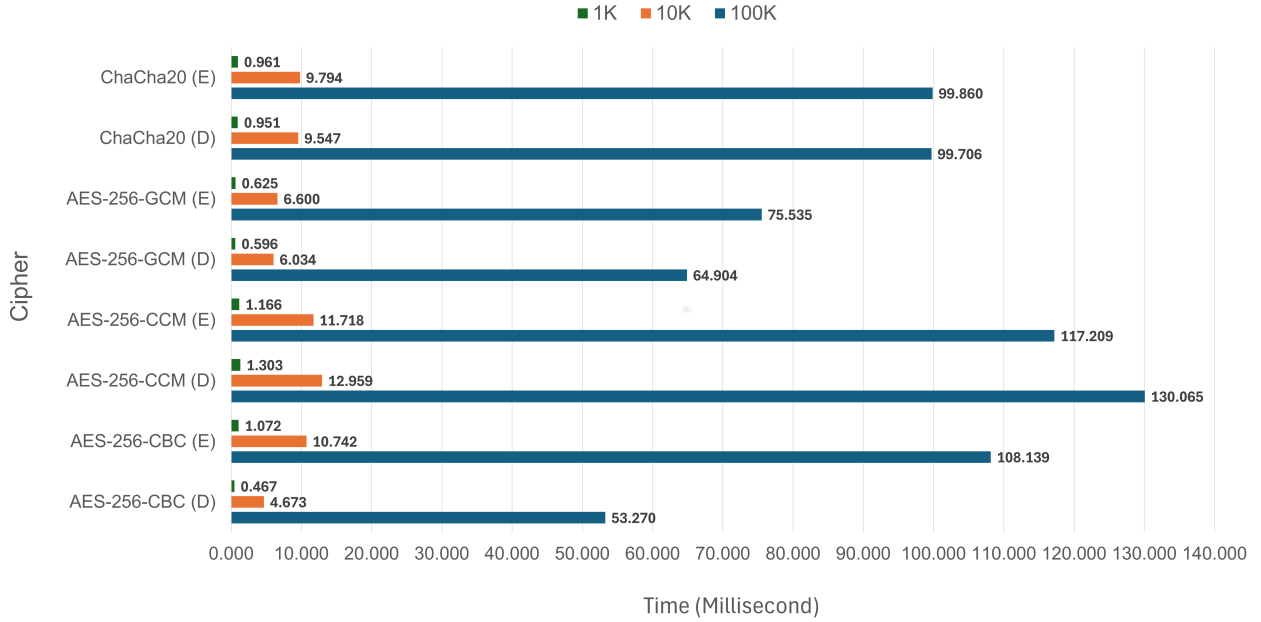


Fig. 5. Symmetric-key ciphers performance for encrypting and decrypting IP packets (E means encryption and D means decryption)

the signaling messages of a routing protocol. We used 1500-byte IP packets, 256-bit ciphers, and conducted the experiment on a standard computer. Our experiment measures the time it takes for the ciphers to encrypt or decrypt 1k, 10k, and 100k IP packets continuously assuming that the entanglement keys are quickly accessible. We run each experiment five times, and the results report the attainable average over all trials.

As the results in Fig. 5 show, nearly all tested ciphers in the experiment can approximately achieve an average sub-millisecond performance for continuous encryption and decryption of one thousands IP packets, which is negligible given that routing messages only contain signaling data used by the routing protocols.

We measured the performance of encryption and decryption tasks independently. A router, however, can perform both tasks simultaneously even with small amounts of memory and CPU capacity. We note that routing messages used by the routing protocols within ASes are variable-length and typically contain fewer than 1500 bytes of routing information, and some protocols to have a maximum message size (e.g., 64KB in OSPF which can approximately fit into fifty IP packets). Therefore, using one thousands IP packets offers a good approximation of the encryption and decryption delay in routers that generate and forward messages to their neighbors in a small to medium-sized network, assuming that encryption and decryption tasks may occur in parallel.

For larger network environments, our results in Fig. 5 show an average of less than 15 milliseconds of cipher delay with routers that continuously encrypt or decrypt ten thousands IP packets. This is still not significant, given that the convergence time for most intra-AS routing protocols is in the order of seconds [22].

As the number of IP packets used for signaling increases to large amounts (e.g., 100 thousand in our experiment), we can expect the average processing time to increase substantially across all ciphers, as shown in Fig. 5. However, the time is still under 150 milliseconds. This amount of IP packets is much more than what the signaling of a routing protocol requires in response to events in the network even in very large settings [22].

In our experiments, we also observed that, as the number of IP packets increases, the cipher delay increases with an approximately linear rate and the overhead of packet decryption is typically less than encryption. However, we did not include the impact of the network interface type (e.g., Wi-Fi or Ethernet), network bandwidth, or the protocol stack processing delay (e.g., encapsulation and de-multiplexing) in our experiments which may affect such balance in the results.

Cipher	Time (Millisecond)		
	1K	10K	100K
Camellia-256 (E)	8.061	79.151	799.759
Camellia-256 (D)	7.464	73.316	741.859

TABLE I
CAMELLIA CIPHER PERFORMANCE FOR ENCRYPTING AND DECRYPTING IP PACKETS (E MEANS ENCRYPTION AND D MEANS DECRYPTION)

In addition to the ciphers shown in Fig. 5, we conducted the same experiment using Camellia [7], which is currently less adopted in various network environments. Unlike the other ciphers in our experiment, Camellia showed a relatively lower performance. We believe this is mainly because of the lack of hardware acceleration support (e.g., AES-NI and ARM SafeXcel). By contrast, the AES ciphers we tested enjoy hardware acceleration. Similar to Camellia, ChaCha20 [19] does not support a native hardware acceleration, but its

performance is comparable to AES. This is because its design avoids using time consuming table lookups.

V. STRONG CONFIDENTIALITY OF CONCISE

This section shows that the likelihood of deciphering CONCISE-protected messages as an outsider is insignificant. Hence, the only viable way to attack a routing protocol protected using CONCISE is to subvert the routers themselves. This is far more challenging than attempting to impersonate trusted routers or inject false routing information in signaling messages sent in plain text, as it is done today.

A. Security Against Brute-Force Attacks

A brute-force attack is one in which adversaries rely on trial and error to find the encryption keys or any useful information that allows them to find the keys eventually. The process can continue until the keys are found. The main defense against these attacks in CONCISE is the secrecy of values and length of integers in set $\mathcal{I}$ from which encryption keys are derived, and for some ciphers the length of encryption keys themselves (L_c). If adversaries manage to somehow discover the key length L_c used in a particular network, they must search through a key space of 2^{L_c} keys, which is very large even for the case of AES using 128 bits for encryption keys.

B. Security Against Known Plain-Text Attacks

In a known plain-text attack, adversaries who have access to both plain text (unencrypted data) and the cipher message (encrypted data) present a target encryption algorithm with the plain text to study the patterns of encryption emerging in the cipher message with the goal of breaking the encryption algorithm. Adversaries can only gain limited insight on the keys used in an entanglement because the sender selects new keys for each message at random and any given encryption key is not reused for a very long time, which is a function of the key length L_c .

C. Security Against Attacks on CONCISE Operation

Adversaries could attempt to attack CONCISE using the way it operates but without knowing the shared secrets used to configure a network. However, they would face insurmountable difficulty attempting such an attack, and would have no better likelihood of success than with a brute-force attack. This is due to the secrecy of key lengths and the way in which encryption keys are derived from a basis key.

When AES is used in CONCISE, the key used to encrypt each message is drawn from integers in set $\mathcal{I}$ and adversaries have no insight as to which integer values of some length L were used as basis keys. This is because basis keys are selected using a random function by the sender and the function uses parameter values that are secret outside an entanglement. A symmetric key corresponding to the 128, 192, or 256 least-significant bits of a basis key k_0 is used for each message being sent in an entanglement. Given that k_0 is chosen randomly by the sender, the probability of adversaries deciphering a ciphertext is the same as in AES using a random symmetric key.

Hence, the probability of successfully deciphering a message using the mechanisms defined in CONCISE using AES without knowing the secrets with which a network is configured is insignificant.

VI. SECURITY FEATURES OF CONCISE

A computer network with trusted routers using CONCISE enjoys ease of configuration, strong confidentiality, integrity, and cryptographic authentication.

Ease of Configuration: This is a major motivation for the design of CONCISE, and is not a security feature in itself. However, the simplicity of configuring trusted routers helps prevent attacks due to misconfigurations or lax configurations. All trusted routers are configured with the same two secrets, rather than requiring different secrets for different network links as it is the case in OSPFv2, for example.

Confidentiality: In contrast to all prior routing security approaches, CONCISE allows trusted routers to exchange signaling messages that are always encrypted, without the need for any handshake to establish trust or disseminate keys, and without the need to use IPsec. As such, it is immune to man-in-the-middle attacks. Current alternatives require trusted routers (e.g., routers) to be configured with static symmetric keys per neighbor, or the use of mathematically complex ways of generating static symmetric keys from public-private key pairs.

As we have shown, CONCISE provides strong confidentiality, and Section IV shows that the cryptographic approach used in CONCISE is very fast.

Integrity: The CONCISE header accompanies a cryptographic hash of the entire header elements with the bits of all the message blocks before they are encrypted independently of whether the selected cipher performs integrity checks. The hash is also accompanied by a timestamp indicating the time when the hash is computed. Using the hash and its timestamp in CONCISE headers provides a strong tool for integrity checks before the message blocks in the payload are decrypted. If message blocks in the message payload are tampered, a trusted receiver drops the packet.

Message Authentication: The hash header element used for integrity check in a CONCISE header is also used for cryptographic message authentication independently of the cipher being used because of the correlated use of encryption keys. If an integrity check is passed, it means that the sender is using the same encryption keys as the receiver. The receiver can verify that the sender is who it claims to be based on the correlated use of entanglement integers. This is much stronger than the authentication methods currently used in routing protocols.

VII. CONCLUSIONS AND FUTURE WORK

We described the use of cryptographic entanglements as an alternative to existing key-distribution methods to secure routing within an AS independently of the routing protocols being used. We presented CONCISE as an approach to implement

cryptographic entanglements assuming that AES is used as the cipher. A major advantage of replacing current approaches for key distribution with CONCISE is that it greatly simplifies the configuration of trusted routers. Furthermore, CONCISE eliminates the vulnerabilities and complexity resulting from having to trust key-distribution servers and protocols, relaying on public-key infrastructures, or using cryptographic handshakes subject to man-in-the-middle attacks.

CONCISE maintains cryptographic entanglements even in the presence of significant packet losses.

The results of our experiment measuring the performance of AES using entanglement-based encryption keys demonstrate that CONCISE can easily support per-hop encryption and decryption and be used to secure any routing protocol without modifying it. This makes CONCISE an ideal approach to provide secure routing in the ASes of the current Internet and the Internet of the future.

Our future work addresses the use of CONCISE to protect routing across ASes and developing security countermeasures to protect against attacks based on subverting routers.

ACKNOWLEDGMENTS

This work was supported by the Canada Excellence Research Chair in Intelligent Digital Infrastructures at the University of Toronto, funded by the Tri-agency Institutional Programs Secretariat.

REFERENCES

- [1] H. Abbas et al., "Security Assessment and Evaluation of VPNs: A Comprehensive Survey," *ACM Computing Surveys*, Dec. 2023.
- [2] C. Adams, "Dictionary Attack," *Encyclopedia Of Cryptography, Security And Privacy*, 2025.
- [3] L. Åkerberg and V. Johansson Baurne, "Practical Analysis and Simulation of OSPF Protocol Vulnerabilities," Tech. Rep., KTH Royal Institute of Technology, Sept. 2024.
- [4] F. A. Alaba et al., "Internet of Things Security: A survey," *Journal of Network and Computer Applications*, Vol. 88, pp. 10–28, 2017.
- [5] R. Albrightson, J.J. Garcia-Luna-Aceves, and J. Boyle, "EIGRP—A Fast Routing Protocol based on Distance Vectors," *Interop '94*, 1994.
- [6] M. Alenezi et al., "Symmetric Encryption Algorithms: Review and Evaluation Study," *International Journal Of Communication Networks And Information Security*, 2020.
- [7] K. Aoki et al., "A 128-bit Block Cipher Suitable for Multiple Platforms—Design and Analysis," *International Workshop On Selected Areas In Cryptography*, 2000.
- [8] J. Aweya, *IP Routing Protocols: Fundamentals and Distance-Vector Routing Protocols*, CRC Press, 2021.
- [9] A. Babakhouya et al., "SDV: A New Approach to Secure Distance Vector Routing Protocols," *Proc. 2006 IEEE SecureCom / SECOVAL Workshop*, 2006.
- [10] A. Barbir, S. Murphy and Y. Yang, "Generic Threats to Routing Protocols," RFC 4593, IETF, Oct. 2006.
- [11] M. Bhatia et al., "OSPFv2 HMAC-SHA Cryptographic Authentication," RFC 5709, Oct. 2009.
- [12] A. Boukerche (Ed.), *Handbook of Algorithms for Wireless Networking and Mobile Computing*, Chapman & Hall/CRC, 2006.
- [13] B. Bruhadeshwar et al., "Routing Protocol Security Using Symmetric Key based Techniques," *Proc. ARES 2009*, 2009.
- [14] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2019.
- [15] A. Chakrabarti and G. Manimaran, "Internet Infrastructure Security: A Taxonomy," *IEEE Network*, Nov. 2002.
- [16] R. Coltun et al., "OSPF for IPv6," RFC 5340, 2008.
- [17] M. Gupta and N. Melam, "Authentication/Confidentiality for OSPFv3," RFC 4542, June 2006.
- [18] E. Demirci and E. Olca, "A Review of Grover's Search Algorithm: Applications and Modifications," *Proc. 9th International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2025.
- [19] F. De Santis, "ChaCha20-Poly1305 Authenticated Encryption for High-Speed Embedded IoT Applications," *Design, Automation & Test In Europe Conference & Exhibition (DATE)*, 2017.
- [20] D. Dinev and G. Spasova, "Benchmarking and Teaching Symmetric Cryptography: A Comparative Study of Algorithms and Cryptanalysis Methods with an Educational Simulator," *Proc. CIEES 2025*, 2025.
- [21] C. Dwork et al., "Exposed! A Survey of Attacks on Private Data," *Annual Review Of Statistics And Its Application*, 2017.
- [22] J.J. Garcia-Luna-Aceves, M. Moghaddassian, and C. Hsieh, "Spreading Computations for Loop-Free Multipath Routing in Computer Networks," *Proc. 2024 IEEE IPCCC*, 2024.
- [23] A. Gheorghies et al., "A Comparative Study of Cryptographic Key Distribution Protocols," *Cryptology EPrint Archive*, 2021.
- [24] H. Gredler and W. Goralski, *The Complete IS-IS Routing Protocol*, Springer, 2005.
- [25] V. Hassija et al., "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.
- [26] A.M. Hegland et al., "A Survey of Key Management in Ad Hoc Networks," *IEEE Communications Surveys & Tutorials*, 2006.
- [27] V.-H. Hoang, E. Lehtihet, and Y. Ghamri-Doudane, "Password-Based Authenticated Key Exchange based on Signcryption for the Internet of Things," *Proc. IEEE 2019 Wireless Days (WD)*, 2019.
- [28] Y. Hu, A. Perrig, and D. Johnson, "Efficient Security Mechanisms for Routing Protocols," *Proc. NDSS '03*, 2003.
- [29] G. Huston, M. Rossi, and G. Armitage, "Securing BGP — A Literature Survey," *IEEE Communications Surveys & Tutorials*, 2011.
- [30] M. Jayakumar, N. Rekha, and B. Bharathi, "A Comparative Study on RIP and OSPF protocols," *2015 ICIIECS*, 2015.
- [31] S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303, 2005.
- [32] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of Things Security: A Top-Down Survey," *Computer Networks*, 2018.
- [33] N. Leavitt, "Internet Security under Attack: The Undermining of Digital Certificates," *Computer*, 2011.
- [34] I. Mansour et al., "Secure Multihop Key Establishment Protocols for Wireless Sensor Networks," *Proc. International Conference on Cryptography and Security Systems*, 2014.
- [35] M. R. Mishra and J. Kar, "A Study on Diffie-Hellman Key Exchange Protocols," *International Journal of Pure and Applied Mathematics*, 2017.
- [36] S. K. Mousavi et al., "Security of Internet of Things based on cryptographic algorithms: A Survey," *Wireless Networks*, 2021.
- [37] J. Moy, "OSPF Version 2," RFC2328, 1998.
- [38] S. Murphy and M. Badger, "Signature Protection of the OSPF Routing Protocol," *Proc. NDSS '96*, 1996.
- [39] S. Murphy et al., "OSPF with Digital Signatures," RFC 2154, 1997.
- [40] G. Nakibour et al., "Persistent OSPF Attacks," *Proc. NDSS*, 2012.
- [41] P. Papadimitratos and Z. Haas, "Securing the Internet Routing Infrastructure," *IEEE Communications Magazine*, 2002.
- [42] D. Sarangi, "A Comparative Study of AES Encryption Modes and Hashing for Blockchain Applications," National College of Ireland, 2024.
- [43] J. Schwenk, "IP security (IPsec)," *Guide To Internet Cryptography: Security Protocols And Real-World Attack Implications*, 2022.
- [44] K.-A. Shim, "A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials*, 2015.
- [45] B. Smith and J.J. Garcia-Luna-Aceves, "Securing Distance Vector Routing Protocols," *Proc. NDSS Symposium*, 1997.
- [46] B. Vetter, F. Wang, and S.F. Wu, "An Experimental Study of Insider Attacks for OSPF Routing Protocol," *Proc. IEEE ICNP '97*, 1997.
- [47] K. Wai, "Analysis of RIP, EIGRP, and OSPF Routing Protocols in a Network," *International Journal Of Trend In Scientific Research And Development*, 2019.
- [48] T. Wan et al., "S-RIP: A Secure Distance Vector Routing Protocol," *Proc. Applied Cryptography and Network Security*, 2004.
- [49] H. Yang et al., "Security in Mobile Ad Hoc Networks: Challenges and Solutions," *IEEE Wireless Communications*, 2004.