Multipath Key Exchange Scheme Based on the Diffie-Hellman Protocol and the Shamir Threshold

Daouda Ahmat[1], Marayi Choroma[2], and Tegawendé F. Bissyandé[3,4]
(Corresponding author: Daouda Ahmat - daouda.ahmat@uvt.td)

Virtual University of Chad[1]
B.P: 5711, General Daoud Soumaine Road, N'Djamena, Chad
University of N'Djamena[2]
B.P: 1117, Mobutu Street, N'Djamena, Chad
University of Luxembourg, SnT[3]
2, avenue de l'Université L-4365 Esch-sur-Alzette, Luxembourg
University Ouaga I Pr Joseph Ki-Zerbo[4]
B.P: 7021, Ouagadougou, Burkina Faso
(Email: daouda.ahmat@uvt.td[1] / choroma.marayi@auf.org[2] / tegawende.bissyande@uni.lu[3,4])
(Received Dec. 18, 2017; Revised and Accepted June 18, 2018; First Online Jan. 13, 2019)

Abstract

In the Internet, as well as in any open autonomous distributed systems, threats to secure communications are pervasive. We contribute towards adressing them by proposing, in this paper, a new multipath key exchange approach, which does not rely on any centrally trusted coordinator. This approach is thus suitable for use in distributed systems such as widespread P2P networks or booming wireless mesh networks (e.g., for the Internet-of-Things). We design a new algorithm based upon an extension of both the Diffie-Hellman protocol and the Shamir threshold scheme. In order to overcome man-in-the-middle attacks inherent to the Diffie-Hellman key exchange model, our proposed approach guarantees secure key exchange by exploring disjoint transmission paths and the Shamir threshold scheme. The public key is used as the root of a polynomial of degree $k-1$, and n points of this polynomial are generated and transmitted from source to destination, each point through a disjoint path. Upon reception of at least k points among n, the receiver is able to reconstruct the complete key. In addition, this paper demonstrates how the disjoint paths constructions and the routing algorithms are designed to work regardless of the network topology.

Keywords: Diffie-Hellman; Key Exchange; Multipath Routing; P2P Networks; Shamir Secret Sharing

1 Introduction

A growing number of security attacks on distributed systems such as the Internet and P2P networks for the Internet-of-things (IoT), have led to an increasing interest in the research community. In this regard, many solutions have been proposed to secure traffic across networks by using a security infrastructure based on a central authority through which s cryptographic keys are dispatched. However, centralized key exchange systems are not suitable for autonomous distributed systems such as peer-to-peer networks. Infrastructure-less key exchange techniques, which are not tied to a central node for key negotiation, and which can be usable over *insecure* networks are thus necessary in the context of such systems.

The end-to-end key exchange scheme proposed by the Diffie-Hellman protocol [11] enables a key exchange between two remote correspondents. However, due to its vulnerability to interception attacks, the Diffie-Hellman protocol cannot be leveraged as-is. Our approach, in this paper, aims at overcoming this issue by combining the Diffie-Hellman protocol with both multipath routing and the Shamir's threshold scheme [32]. Concretely, the Shamir's secret sharing enables us to divide a key $\mathcal{K}$ into n subkeys in such a way that $\mathcal{K}$ is reconstructable from any k subkeys. A security property in this scheme is that any subset of up to $k-1$ subkeys cannot leak information about $\mathcal{K}$.

The contributions of the paper are exposed in the re-

mainder of the paper, which is structure as follows:

- We first discuss the scope and features provided by literature work, then we enumerate the limitations of our own previous work towards implementing secure network exchanges (cf. Section 2).
- We then present background information related to previous work (cf. Section 3), before providing detailed descriptions on the design of our key exchange scheme (cf. Section 4), including the implementation of several multipath routing approaches (cf. Section 5).
- Finally, we analyse the security advantages of our approach (cf. Section 6) and discuss experimental results —based on network simulations— that assess the efficiency of the proposed approach (cf. Section 7).

2 Related Work

A number of research works have presented various security infrastructures over fully decentralized or ad hoc networks [5,10,13,23,35,41]. Although they are designed to be suitable in such environments, the proposed approaches come with different caveats. In this section, we describe some models from the literature and highlight the potential benefits of our approach.

Srivasta and Liu have relied on the Diffie-Hellman algorithm to deliver a solution that prevents threats in DHT networks [34]. Wang *et al.* have built a distributed PKI on top of the Chord structured overlay network [2]. They have used threshold cryptography to distribute the functionality of the PKI across the nodes of the DHT network. This Chord-PKI provides traditional PKI features such as certification, revocation, storage and retrieval.

The literature now includes a number of approaches [8, 9,12,15,17,22,24,26–30,40] that extend the Diffie-Hellman key exchange algorithm. Nevertheless, there are scarce works which address end-to-end key exchange problem based on both distributed systems such as peer-to-peer networks by leveraging the Diffie-Hellman protocol and proposing multipath subkeys routing and multi-secret mechanisms. Takano *et al.* [35] have investigated this avenue over a decade ago. This approach is based on ring topology and does not provide explanation about its key splitting technique.

Jiejun *et al.* propose to distribute certification authority functions through a threshold secret sharing mechanism [21]. In this system, the private key is computed by k neighbor nodes and the public key is derived from node identity.

Many threshold schemes have been directly derived from traditional Shamir threshold [32] to address multi-secret sharing mechanisms. Among them we can quickly cite Brian King [20], Appala *et al.* [36], Harsha *et al.* [19], Rao *et al.* [39], Yang *et al.* [42], Guo *et al.* [18], Ting-Yi Chang *et al.* [38], Ting-Yi Chang [7] and Chang *et al.* [6]. We will focus our study on the mainstream protocol known under name of the *Shamir threshold* [32].

Fathimal *et al.* [16] recently proposed an extension of Shamir's method that enables to retrieve a key from p subkeys, where $p \leq k-1$. p is a threshold number of equally-weighted from each compartment.

Threshold cryptography is also used in identity-based key management [10]. The main idea for identity-based cryptography is to define public keys derived from the identities of communicating nodes [33]. Unfortunately, node identity updates lead to frequent key changes.

Myrmic [41] is a DHT-based system that proposes a secure and robust routing protocol. Designed to be robust against adversarial interference, Myrmic introduces the concept of *Neighborhood Authority* in order to handle certificates in a small set of nodes.

Takano *et al.* have designed a Multipath Key Exchange [35] similar to that proposed in our work. Their technique however was designed to fit the Symphony and Chord P2P systems that are both based upon a ring topology. Their proposed approach, based on probabilistic clockwise/anticlockwise routing, is thus sensitive to coordinated MITM attacks by two attackers.

Jaydip Sen proposes a multipath certification protocol for MANETs that proceeds by broadcasting in order to discover the route between both source and destination nodes [31]. The key exchange protocol is based on this routing approach to retrieve the public keys of the nodes. However, broadcasting techniques have proven to not be relevant for large scale networks such as fully decentralized P2P systems.

El Hajj Shehadeh *et al.* investigate secret key generation from wireless multipath channels [13]. The proposed protocol is based mainly on both the physical characteristics of the wireless channel and a key pre-distribution scheme. This solution is implemented within the physical layer and does not scale to large networks.

3 Background

In previous work [1], we have proposed a key exchange scheme based on an extension of the Diffie-Hellman protocol. Our approach enabled sharing a secret by using multiple disjoint paths in a P2P system called CLOAK [4,37]. The scheme was mainly devised to overcome the vulnerability of the Diffie-Hellman protocol to Man-In-The-Middle (MITM) attacks.

In this scheme, we would split the public key into several subkeys that would then be sent over several disjoint paths to the destination. The destination needed to recover all subkeys in order to get the public key of the sender. A major issue with this approach was that the interception of most (*i.e.*, not even all) of the subkeys by an attacker, allows him to make a brute force attack on the missing subkey(s) which are smaller than the original key. Indeed, if the subkeys set is $S^K = \{0,1\}^*$, where $|S^K| = \rho$, interception of α key components among n by

an attacker, with $\alpha \leq n$, reduces the difficulty to carry out a brute force attack to $\frac{\rho-\alpha\frac{\rho}{n}}{n}$, that is $\rho\frac{n-\alpha}{n^2}$. In addition, in our previous work, the proposed multipath routing algorithm for the subkeys was exclusively suitable to our CLOAK P2P system [37], making its exploitation challenging in other systems.

In this work, we try to address the issues and limitations of this prior work by integrating the Shamir's shared secret scheme in our previous solution.

4 Scheme Design

In this section, we describe our key exchange scheme and its corollary properties which are suitable to distributed networks, *i.e.*, networks that lack any trusted central coordination point.

4.1 Diffie-Hellman Vulnerability

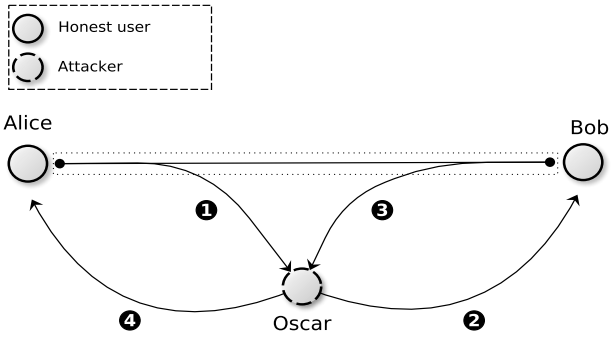

Figure 1: Man-in-the-middle attack

The Diffie-Hellman protocol is an algorithm initiated by two distant correspondents that cooperate to remotely accomplish key exchange tasks. As shown in Figure 1, one fundamental problem of the Diffie-Hellman protocol is its vulnerability to interception attacks, known as Man-In-The-Middle (MITM) attacks. This figure displays a scenario where an attacker, *Oscar*, eavesdrops the channel used by both *Alice* and *Bob* to exchange cryptographic data. ❶ and ❸ flows represent intercepted data by the attacker *Oscar*, data transmitted respectively by *Alice* and *Bob*. ❷ and ❹ show corrupted flows produced by *Oscar* and then respectively transmitted to *Bob* and *Alice*. Thus, *Oscar* can perpetrate attacks: it can eavesdrop, replay or modify data exchanged between *Alice* and?*Bob*. Specifically to the Diffie-Hellman protocol, *Oscar* can intercept the public key sent by *Alice* and send its own public key to *Bob* and can do the same in the other direction with another public key generated to replace the public key sent by *Bob*.

4.2 Protocol Overview

Based upon multipath routing, our approach aims at reducing the Diffie-Hellman vulnerability by combining the two cryptographic algorithms mentioned earlier. Hence, key $\mathcal{K} = g^s \bmod (p)$ must be splitted into n subkeys $s_{k_0}, \ldots, s_{k_{n-1}}$ and then each subkey s_{k_i} will be subsequently sent through a disjoint path. Shamir's threshold algorithm is applied in both splitting and reconstruction of key $\mathcal{K}$. In addition, various routing techniques are proposed in order to route subkeys over network through disjoint paths.

4.3 Key Management

Our key exchange approach is based upon both the Diffie-Hellman protocol and Shamir's threshold. We now describe the mechanisms behind our key exchange scheme.

4.3.1 Key Splitting

In order to forge subkeys, each correspondent firstly creates a secret key $S^K = s$ and generates a polynom $f^l(X)$, where $f^l(0) = g^s (\bmod\ p)$, as shown in Algorithm 1. Then, for each $x_{i\ 0\leq i\leq n}$, with $x_i \neq 0$, $f^l(x_i)$ is computed. Finally, all interpolation points $(x_i, f^l(x_i))$, except $(0, f^l(0))$, are stocked in `subKeysList`. Algorithm 2, which depends on Algorithm 1, provides more details about the key splitting scheme.

Algorithm 1: Creation of polynom of degree k

`createPolynom`(k, $g^S (\bmod\ p)$) **return** Polynom;
begin
  $a_0 \leftarrow g^S (\bmod\ p)$;
  $f^l(0) \leftarrow a_0$;
  $i \leftarrow 1$;
  while $i \leq k-1$ **do**
    $a_i \leftarrow$ `getRandomCoefficient`();
    if $i = k-1$ and $a_i = 0$ **then**
      continue;
    $l_i(X) \leftarrow a_i X^i$;
    $i \leftarrow i+1$;
  $f^l(X) \leftarrow \sum_{i=1}^{k-1} l_i(X) + f^l(0)$;
  return $f^l(X)$;

4.3.2 Key Reconstitution

On receiving of $(x_i, f^l(x_i))_{0\leq i\leq n}$, correspondent node applies Algorithm 3 in order to rebuild key $\mathcal{K} = g^s (\bmod\ p)$ from received subkeys $s_{k_0}, \ldots, s_{k_{p-1}}$, that are equivalent to $(x_i, f^l(x_i))_{0\leq i\leq p}$, where $p \geq k (=$ `degree of` $f^l)$. Indeed, receiver computes $s_{k_0} \odot s_{k_1} \odot \cdots \odot s_{k_{p-1}} = g^s (\bmod\ p) = \mathcal{F}^l(0)$, such that:

Algorithm 2: Subkeys generation

Input: $k, n,\ g^S(\text{mod } p)$
$f^l(X) \leftarrow \texttt{createPolynom}\left(k,\ g^S(\text{mod } p)\right)$;
subKeysList $\leftarrow \perp$;
begin
  $i \leftarrow 1$;
  while $i \leq n$ **do**
    $x_i \leftarrow \texttt{getRandomValue}()$;
    $\widehat{f}(x_i) \leftarrow \left(x_i, f^l(x_i)\right)$;
    $\texttt{storeInSubKeysList}\left(\widehat{f}(x_i), \text{subKeysList}[i]\right)$;
    $i \leftarrow i + 1$;

$$\mathcal{F}^l(X) = \sum_{j=0}^{p} y_j l_j(X) \tag{1}$$

where $y_i = f^l(x_i)$ and

$$l_j(X) = \prod_{i=0, i\neq j}^{p} \frac{X - x_i}{x_j - x_i} \tag{2}$$

Algorithm 3: Reconstitution of key from received subkeys

Input: $k, (x_i, y_i)_{0\leq i\leq n}$
Output: Key
begin
  if $|(x_i, y_i)_{0\leq i\leq n}| < k$ **then**
    return $\perp$;
  else
    foreach $i \in [\![0, n]\!]$ **do**
      $l_i(X) \leftarrow \prod_{j=0, j\neq i}^{n} \frac{X - x_j}{x_i - x_j}$;
    $j \leftarrow 0$;
    $f_l(X) \leftarrow 0$;
    while $j \leq n$ **do**
      $f_l(X) \leftarrow y_j \times l_j(X) + f_l(X)$;
      $j \leftarrow j + 1$;
    return $f_l(0)$;

4.4 Key Exchange Protocol

Algorithm 4 summarizes the process of our key exchange approach: the Diffie-Hellman protocol is relied upon firstly to generate a key of shape $\mathcal{K} = g^s(\text{mod } p)$; then Shamir's threshold is leveraged to split the key into several subkeys or to rebuild the key from its component subkeys $s_{k_0}, s_{k_1}, \cdots, s_{k_n}$. Precisely, equations 1 and 2 describe *Lagrange Interpolation* used in order to rebuild key $\mathcal{K}$ original.

Algorithm 4: Multipath key exchange protocol

public data:	**private data:**
$\boldsymbol{p}$: a prime number	$\mathbf{s}_a$: secret key of **Alice**
$\boldsymbol{g}$: a generator	$\mathbf{s}_b$: secrete key of **Bob**

1) **Alice** creates s_a and she then computes her partial key $\text{Key}_a = g^{s_a} \text{ mod } (p)$;
2) **Alice** generates a polynom f_a^l of degree k, such as $f_a^l(0) = \text{Key}_a$, and she then computes n interpolation points of the polynom: $\widehat{f}_{a_0}, ..., \widehat{f}_{a_{n-1}}$ (where $n \geq k$);
3) **Alice** sends n subkeys $\widehat{f}_{a_i}$, except $(0, f^l(0))$ point, to **Bob** via disjoint paths;
4) **Bob** determines $L^b(X)$ according to $\widehat{f}_{a_i}$, received from **Alice**, and subsequently computes $L^b(0)$ which gives $g^{s_a} \text{ mod } (p)$, if $|\widehat{f}_{a_i}| \geq k$;
5) **Bob** creates s_b and he then generates his partial $\text{Key}_b = g^{s_b} \text{ mod } (p)$;
6) **Bob** forges a polynom f_b^l, such as $f_b^l(0) = \text{Key}_b$, and he then determines n interpolation points of the polynom $\widehat{f}_{b_0}, ..., \widehat{f}_{b_{p-1}}$ (where $p \geq k$);
7) **Bob** sends n subkeys $\widehat{f}_{b_i}$, except $(0, f^l(0))$ point, to **Alice** through disjoint paths;
8) **Alice** generates $L^a(X)$ from $\widehat{f}_{b_i}$ received from **Bob** and she then computes $L^a(0)$ which gives $g^{s_b} \text{ mod } (p)$, if $|\widehat{f}_{b_i}| \geq k$;
9) **Alice** computes Key $= \text{Key}_a \times L(0) = g^{s_a} \text{ mod } (p) \times g^{s_b} \text{ mod } (p) = g^{s_a s_b} \text{ mod } (p)$;
10) **Bob** computes Key $= \text{Key}_b \times L(0) = g^{s_b} \text{ mod } (b) \times g^{s_a} \text{ mod } (p) = g^{s_b s_a} \text{ mod } (p)$;

5 Multipath Routing Policy

In this section, we provide technical details about multipath routing algorithms and then point out their performance differences.

5.1 Deterministic Routing: Pre-routing and Then Routing

Deterministic routing: Enables to route subkeys through disjoint and predetermined paths, as described in Algorithm 5. In other words, each subkey is sent via a disjoint path whose constituting hops are all determined in advance. *Deterministic routing* is however not suitable for dynamic environments where topologies change constantly.

5.2 Non-deterministic Routing: Both Marking and Routing

Non-deterministic routing: Detailed in Algorithm 6, enables to route each subkey through a disjoint path, but unlike deterministic routing, determines on the fly the hops that form each disjoint path.

Algorithm 5: Deterministic routing

Input: $G = (V, E), (s, t), \mathsf{SubKeysList}$
$\mathcal{V} \leftarrow \emptyset$;
begin
  $k \leftarrow |\mathsf{SubKeysList}|$;
  $j \leftarrow k - 1$;
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{s, t\}$;
  while $j \geq 0$ **do**
    $node \leftarrow s$;
    while $node \neq t$ **do**
      $d_{min} \leftarrow \texttt{distance}(node, t)$;
      foreach $neighbor \in \texttt{neighborsListOf}(node)$ **do**
        $d \leftarrow \texttt{distance}(neighbor, t)$;
        if $neighbor \notin \mathcal{V}$ and $d < d_{min}$ **then**
          $d_{min} \leftarrow d$;
          $node \leftarrow neighbor$;
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{node\}$;
      $\mathcal{P}_j \leftarrow \mathcal{P}_j \cup \{node\}$;
    $j \leftarrow j - 1$;
  while $k > 0$ **do**
    $e \leftarrow \mathsf{SubKeysList}[k]$;
    $\texttt{sendViaPath}(e, \mathcal{P}_k)$;
    $k \leftarrow k - 1$;

Algorithm 6: Non-deterministic routing

Input: $G = (V, E), (s, t), \mathsf{SubKeysList}$
$\mathcal{V} \leftarrow \emptyset$;
begin
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{s, t\}$;
  foreach $e \in \mathsf{SubKeysList}$ **do**
    $node \leftarrow s$;
    while $node \neq t$ **do**
      $d_{min} \leftarrow (node, t)$;
      foreach $neighbor \in \texttt{neighborListOf}(node)$ **do**
        $d \leftarrow \texttt{distance}(neighbor, t)$;
        if $neighbor \notin \mathcal{V}$ and $d < d_{min}$ **then**
          $d_{min} \leftarrow d$;
          $node \leftarrow neighbor$;
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{node\}$;
      $\texttt{forward}(e, node)$;

5.3 Technical Comparison Between Routing Algorithms

Table 1 presents a technical comparison of both performance metrics and features provided by various multipath routing algorithms.

6 Security Analysis

In a multipath key exchange scheme, a malicious node that wishes to compromise a key being exchanged must be able to collect each of all key components routed over the network. Formally, when paths $\mathcal{P}_0, ..., \mathcal{P}_{k-1}$ are used to send several distinct subkeys from source $\mathcal{S}$ to destination $\mathcal{D}$, the only malicious nodes that could compromise the key should be located at the intersection of all paths. In other words, all the malicious nodes belong to a set $M = \bigcap_{i=0}^{k} \mathcal{P}_i$ which represents the set of intersection points of all paths $\mathcal{P}_i$. $\mathcal{S}$ and $\mathcal{D}$ are obviously ignored in this set. Thus, when $\bigcap_{i=0}^{k} \mathcal{P}_i = \emptyset$ (*bigon criterion* is respected [14, Lemma 2.5]), then all paths are disjoint and any MITM attack attempt cannot succeed. In such a desirable case, there exists a k-connected subgrah between $\mathcal{S}$ and $\mathcal{D}$ in the network topology. When $|\bigcap_{i=0}^{k} \mathcal{P}_i| \geq 1$, there exists a real risk that MITM attacks could be committed on exchange transmitted between $\mathcal{S}$ and $\mathcal{D}$. That means that there exists at least one articulation point. Algorithm 7 enables to detect articulation points within network.

Consequently, the probability to have a MITM attack is estimated by $\sigma = \frac{|\bigcap_{i=0}^{k} \mathcal{P}_i|}{|\bigcup_{i=0}^{k} \mathcal{P}_i|}$ (where each path $\mathcal{P}_i$ is constituted of a set of consecutive hops from source $\mathcal{S}$ to destination $\mathcal{D}$). When all used paths are pairwise disjoint, the probability of *isolated* MITM attack (no *coordinated* MITM attack) is then: $\sigma = 0$ (i.e $|\bigcap_{i=0}^{k} \mathcal{P}_i| = 0$).

The number of distinct paths is also dependant on the source node's degree. Thus, for a given *q-regular tree*, if q is a large number, then there is a probability to have several disjoint transmission channels. Nonetheless, despite the robustness of our multipath negotiation approach, cooperative (*i.e.*, coordinated) MITM attacks, where several nodes maliciously cooperate to compromise a key, are possible. However, it is very hard, and excessively costly to launch such an attack in a real environment, especially in distributed systems where network topology changes dynamically. In addition, the key exchange scheme is suitable for P2P networks and designed regardless of a specific network architecture.

In order to improve performance, re-authentication feature is introduced. However, the challenge message used in this phase could be replayed. Furthermore, when a malicious node caches a challenge message, it can then create its copies and send them successively to target node. Thus, target node tries to resolves each challenge request because it does not know which packet is more fresh than the other. Consequently, it will be rapidly saturated with requests from malicious nodes. Therefore, this causes a Denial-of-Service (DoS) attack.

In order to avoid such an attack from malicious nodes,

Table 1: Comparison of Multipath routing strategies

Finding disjoint paths	Complexity and various features				
	Time complexity	Space complexity	Parity [a]	Robustness [b]	Overview [c]
Indeterministic routing	$O(k(\|E\| + \|V\| \log \|V\|))$	$O(k\|V\|)$		$\surd$	
Deterministic routing	$O(k(\|E\| + \|V\|(1 + \log \|V\|)))$	$O(k\|V\|)$	$\surd$		
Menger's theorem	NP	–	–		$\surd$

[a] Parity between the number of disjoint paths and the number of generated subkeys
[b] Resilience to topology change
[c] Knowledge of topology is needed

Table 2: technical comparison of key exchange schemes

Key Exchange Method	Diffie-Hellman	Takano *et al.*	Our model
Robust to MITM		$\surd$	$\surd$
Unpredictable paths	– [b]		$\surd$
Free of particular topology	$\surd$		$\surd$
CMITM[a] implementation	easy	easy	hard
Required subkeys among n	–	n	$k \leq n$
Subkeys robustness level	–	medium	high
|Set of disjoint paths|	$= 0$	> 2	> 2
Topology maintaining cost	$\mathcal{O}(0)$	$\mathcal{O}(log^2 n)$	$\mathcal{O}(0)$

[a] Coordinated MITM attacks
[b] It depends to the knowledge or not of network topology

a timestamp is assigned to each encrypted challenge message. Thus, the target node could distinguish between fresh packets and replayed packets.

Furthermore, during the key negotiation phase, all packets are exchanged in a clear text mode. Thus, traffic analysis attacks could reveal details about captured packets such as *sequence number* or *payload* which is nothing other than the transported subkey. Hence, multipath key exchange is needed to prevent the knowledge of all subkeys.

Table 2 summarizes security and technical features of traditional security protocol, called Diffie-Hellman algorithm [11], key exchange scheme proposed by Takano *et. al* [35] and our key management scheme. This table shows that our scheme is more advanced than other models in several aspects.

Isolated attacks launched over the network cannot compromise multipath key exchange if there are at least two disjoint paths found between two correspondent nodes. However, coordinated attacks launched from various malicious nodes could be potentially able to compromise key by intercepting all its subkeys sent through disjoint paths.

Otherwise, in the new scheme that is proposed in this paper, missing a few of the subkeys, during their transport, does not always cause key exchange failure. Technically, if the number of received keys is greater or equal to the threshold k, with $k \leq n$, then the original key could be reconstructed. Formally, the assertion can take the

Algorithm 7: Articulation point detection

```
getArticulationPoint(s, t) return node;
begin
    i ← j ← 0;
    neighborsList ← neighborsListOf(s);
    foreach neighbor ∈ neighborsList do
        P_j ← ∅;
        createPath(neighbor, t, P_j);
        node ← P_j.getLastNode();
        if node ∉ t then
            i ← i + 1;
        P_j ← P_j ∪ {node};
        j ← j + 1;
    return (i = j) ? ⋂_{i=0}^{n} P_i : ⊥;
```

form of the following theorem.

Theorem 1. *In Shamir's $(k,n)-$threshold scheme, any subset of up to $q = k-1$ subkeys, where $k \leq n$, does not leak any information on the shared secret $\mathcal{K}$.*

Proof. To retrieve key $\mathcal{K}$ from (x_i, y_i) which are employed in Equation (1), let's proceed as follow:

$$\mathcal{K} = \mathcal{F}^l(0) = \sum_{j=0}^{p} y_j \prod_{i=0, i\neq j}^{p} \frac{-x_i}{x_j - x_i} \tag{3}$$

where $k \leq p \leq n$.

Given $|(x_i, y_i)|_{1\leq i\leq q}$, with $q < k \Rightarrow q < p$. Let's suppose that $p = k$, that means that Equation (3) becomes in developped form:

$$\mathcal{K} = \sum_{j=0}^{q} y_j \prod_{\substack{i=0 \\ i\neq j}}^{q} \frac{-x_i}{x_j - x_i} + \sum_{j=q+1}^{k} y_j \prod_{\substack{i=q+1 \\ i\neq j}}^{k} \frac{-x_i}{x_j - x_i} \tag{4}$$

In Equation (4) let's put:

$$\sum_{j=0}^{q} y_j \prod_{\substack{i=0 \\ i\neq j}}^{q} \frac{-x_i}{x_j - x_i} = \mathcal{K}_0 \tag{5}$$

and

$$\sum_{j=q+1}^{k} y_j \prod_{\substack{i=q+1 \\ i\neq j}}^{k} \frac{-x_i}{x_j - x_i} = \mathcal{K}_1 \tag{6}$$

Therefore Equation (4) becomes:

$$\mathcal{K} = \mathcal{K}_0 + \mathcal{K}_1 \tag{7}$$

The value of $\mathcal{K}_1$ is indeterminate because points $(x_i, y_i)_{q+1\leq i\leq k}$ are unknown. That implies $\mathcal{K}$ indeterminate. □

The public key cryptography $g^{s_{k_i}}$ can be published and used in order to verify authenticity of each subkey s_{k_i}. However, public key mechanism requires the use of a traditional centralized public key infrastructure that is incongruous to distributed systems such as peer-to-peer networks.

7 Protocol Assessment

We rely on the Erdõs-Rényi and the Magoni-Pansiot [25] models to build a synthesized graph that represents a random topology. To assess our approach, we use the *nem* simulator[1].

[1] http://www.labri.fr/perso/magoni/nem/

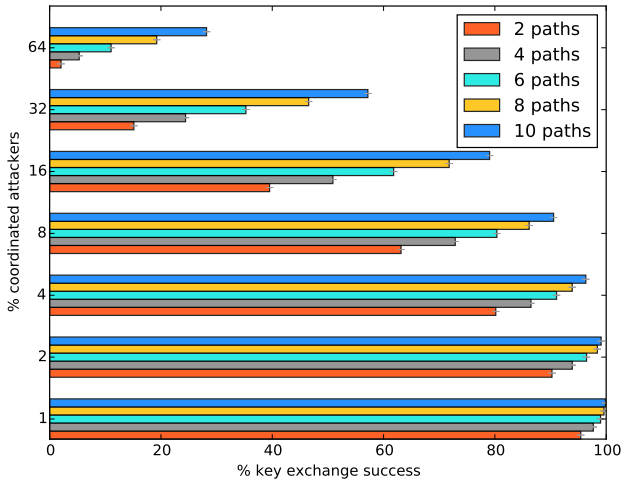

Figure 2: Average number for 10 assessment rounds of key exchange success with respect to both various numbers of disjoint paths found and percentages of coordinated attackers existing over the network

We have carried out the experiments through simulations. We succinctly present the steps that are carried out for the experiments:

- *Definition of the network:* A P2P network is first created. In this step, we set the type of the topology (real map, synthesized topology such as Erdős-Reńyi, Internet-like, etc.) and the size of this network.
- *Selection of the set of compromised nodes:* In the second step, we select a subset of X% nodes which will act as attackers. These nodes are supposed to coordinate their actions.
- *Identification of source and destination nodes:* We then select, among the non-compromised nodes, a pair of source and destination nodes for the data exchange.
- *Data packet transfer:* We launch the transmission by transferring a data packet through the shortest path towards the destination node. All intermediate nodes will be marked and may not be used for another packet between the same pair of source/destination nodes.
- *Check for attacks:* At the end of the packet transfer, we check whether the packet was intercepted by an attacker.
- *Use of alternative paths:* At this time, we start over from step 4, using the same source node but a different path to reach the same destination.
- *Confirmation of the validity of the generated key:* We check whether the packet was potentially intercepted on all disjoint paths. If this is the case, then this

attempt to generate a key is a failure. It is a success otherwise.

- *Change of source/destination nodes:* We repeat the experiments starting from step 3 with a new pair of source and destination nodes.
- *Change of compromised nodes set:* We repeat the experiments starting from step 2 with a new subset of compromised nodes. Basically, we change the percentage of attackers.
- *Change of network settings:* We start over the experiments from step 1 with a new network topology and/or a new size value for the network.

Assessment results are depicted in Figure 2. On the one hand, and despite coordinated attacks, the results show that the higher the number of the disjoint paths, the greater the success rate. On the other hand, the assessment results show also that the higher the rate of attackers within the network, the less the success rate.

8 Conclusion

Currently, security threats in large scale autonomous P2P systems are increasingly present. Given that traditional security protocols fail to be applied in these systems free of central coordination points, we have proposed in this paper a new key exchange algorithm suitable to distributed systems.

It is not only about designing an interesting approach, it is also about a robust scheme. Indeed, the robustness goal is fulfilled by using multipath key exchange technique that extends both Diffie-Hellman protocol and Shamir's threshold in order to meet security expectations. In addition, based on disjoint paths and defined in order to route separately subkeys through the network, multipath routing methods are quite similar to Menger's theorem [3]. Finally, experiments show that our multipath key exchange scheme is robust to isolated MITM attacks and reduces substantially vulnerabilities to distributed MITM attacks as the number of disjoint paths increases.

References

[1] D. Ahmat, D. Magoni, and T. Bissyandé, "End-to-end key exchange through disjoint paths in P2P networks," in *European Alliance for Innovation, Endorsed Transaction on Security and Safety*, pp. 1–15, vol. 2-3, Jan. 2015.

[2] A. Avramidis, P. Kotzanikolaou, C. Douligeris, and M. Burmester, "Chord-pki: A distributed trust infrastructure based on p2p networks," *Computer Networks*, vol. 56, pp. 378–398, Jan. 2012.

[3] T. Böhme, F. Göring, and J. Harant, "Menger's theorem," *Journal of Graph Theory*, vol. 37, no. 1, pp. 35–36, 2001.

[4] C. Cassagnes, T. Tiendrebeogo, D. Bromberg, and D. Magoni, "Overlay addressing and routing system based on hyperbolic geometry," in *Proceedings of the 16th IEEE Symposium on Computers and Communications*, pp. 294–301, 2011.

[5] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *SIGOPS Operating Systems Review*, vol. 36, pp. 299–314, Dec. 2002.

[6] T. Y. Chang, M. S. Hwang, and W. P. Yang, "An improved multi-stage secret sharing scheme based on the factorization problem," *Information Technology and Control*, vol. 40, no. 3, pp. 246–251, 2011.

[7] T. Y. Chang, M. S. Hwang, and W. P. Yang, "An improvement on the lin-wu (t,n) threshold verifiable multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 163, no. 1, pp. 169–178, Apr. 2005.

[8] R. Chaubey and V. R. R. Manthena, *Decryption of Secure Sockets Layer Sessions Having Enabled Perfect Forward Secrecy Using a Diffie-Hellman Key Exchange*, Feb. 13, 2018. US Patent 9,893,883.

[9] S. F. Chiou, M. S. Hwang, and S. K. Chong, "A simple and secure key agreement protocol to integrate a key distribution procedure into the dss," *International Journal of Advancements in Computing Technology*, vol. 4, no. 19, pp. 529–535, Octo. 2012.

[10] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, pp. 107, 2004.

[11] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[12] N. Döttling and S. Garg, "Identity-based encryption from the diffie-hellman assumption," in *Annual International Cryptology Conference*, pp. 537–569, 2017.

[13] Y. E. H. Shehadeh, O. Alfandi, and D. Hogrefe, "Towards robust key extraction from multipath wireless channels," *Journal of Communications and Networks*, vol. 14, no. 4, pp. 385–395, 2012.

[14] D. B. A. Epstein, "Curves on 2-manifolds and isotopies," in *Acta Math*, pp. 15–16, 1966.

[15] A. Escala, G. Herold, E. Kiltz, C. Rafols, and Jorge Villar, "An algebraic framework for diffie–hellman assumptions," *Journal of Cryptology*, vol. 30, no. 1, pp. 242–288, 2017.

[16] P. M. Fathimal and P. A. J. Rani, "Threshold secret sharing scheme for compartmented access structures," *International Journal of Information Security and Privacy*, vol. 10, no. 3, p. 9, 2016.

[17] C. E. Gero, J. N. Shapiro, and D. J. Burd, *Providing Forward Secrecy in a Terminating ssl/tls Connection Proxy Using Ephemeral Diffie-Hellman Key Exchange*, Dec. 27 2016. US Patent 9,531,685.

[18] C. Guo and C. C. Chan, "A novel threshold conference-key agreement protocol based on generalized chinese remainder theorem," *International Journal of Network Security*, vol. 17, no. 2, pp. 165–173, Mar. 2015.

[19] P. Harsha, P. Chanakya, and V. C. Venkaiah, "A reusable multipartite secret sharing scheme based on superincreasing sequence," *International Journal of Network Security*, vol. 20, no. 3, pp. 527–535, May 2018.

[20] B. King, "A dynamic threshold decryption scheme using bilinear pairings," *International Journal of Network Security*, vol. 17, no. 6, pp. 771–778, Nov. 2015.

[21] J. Kong, Z. Petros, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *Ninth International Conference on Network Protocols*, pp. 251–260, 2001.

[22] B. Koziel, A. Jalali, R. Azarderakhsh, D. Jao, and M. Mozaffari-Kermani, "Neon-sidh: Efficient implementation of supersingular isogeny diffie-hellman key exchange protocol on arm," in *International Conference on Cryptology and Network Security*, pp. 88–103, 2016.

[23] H. Kwon, S. Koh, J. Nah, and J. Jang, "The secure routing mechanism for dht-based overlay network," in *10th International Conference on Advanced Communication Technology (ICACT'08)*, vol. 2, pp. 1300–1303, 2008.

[24] J. Liu and J. Li, "A better improvement on the integrated diffie-hellman-dsa key agreement protocol," *International Journal of Network Security*, vol. 11, no. 2, pp. 114–117, Sep. 2010.

[25] D. Magoni and J. J. Pansiot, "Internet topology modeler based on map sampling," in *Proceedings of the 7th IEEE Symposium on Computers and Communications*, pp. 1021–1027, 2002.

[26] T. Mefenza and D. Vergnaud, "Polynomial interpolation of the generalized diffie–hellman and naor–reingold functions," *Designs, Codes and Cryptography*, pp. 1–11, 2018.

[27] H. T. Pan, J. R. Sun, and M. S. Hwang, "Cryptanalysis of biswas's multi-party keys scheme based on the diffie-hellman technique," *International Conference on Advances in Mechanical Engineering and Industrial Informatics*, Jan. 2015.

[28] H. K. Pathak and M. Sanghi, "Simple three party key exchange protocols via twin diffie-hellman problem," *International Journal of Network Security*, vol. 15, no. 4, pp. 256–264, July 2013.

[29] Q. Peng and Y. Tian, "A publicly verifiable secret sharing scheme based on multilinear diffie-hellman assumption," *International Journal of Network Security*, vol. 18, no. 6, pp. 1192–1200, Nov. 2016.

[30] C. Rajarama, J. N. Sugatoor, and T. Y. Swamy, "Diffie-hellman type key exchange, elgamal like encryption/decryption and proxy re-encryption using circulant matrices," *International Journal of Network Security*, vol. 20, 2018.

[31] J. Sen, "A multi-path certification protocol for mobile ad hoc networks," in *The 4th International Conference on Computers and Devices for Communication*, pp. 1–4, 2009.

[32] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612–613, Nov. 1979.

[33] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of Crypto 84 on Advances in Cryptology*, pp. 47–53, 1984.

[34] M. Srivatsa and L. Liu, "Vulnerabilities and security threats in structured overlay networks: a quantitative analysis," in *The 20th Annual Computer Security Applications Conference*, pp. 252–261, 2004.

[35] Y. Takano, N. Isozaki, and Y. Shinoda, "Multipath key exchange on p2p networks," in *The First International Conference on Availability, Reliability and Security*, pp. 8, 2006.

[36] A. N. Tentu, V. K. Prasad, and V. C. Venkaiah, "Secret sharing schemes for multipartite access structures," *International Journal of Applied Engineering Research*, vol. 11, no. 7, pp. 5244–5249, 2016.

[37] T. Tiendrebeogo, D. Ahmat, D. Magoni, and O Sié, "Virtual connections in p2p overlays with dht-based name to address resolution," *International Journal on Advances in Internet Technology*, vol. 5, no. 1, pp. 11–25, 2012.

[38] M. S. Hwang, T. Y. Chang and W. P. Yang, "An improved multi-stage secret sharing scheme based on the factorization problem," *Information Technology and Control*, vol. 40, no. 3, pp. 246 – 251, 2010.

[39] R. Y. V. Subba and C. Bhagvati, "Crt based threshold multi secret sharing scheme," *International Journal of Network Security*, vol. 16, no. 4, pp. 249–255, July 2014.

[40] L. Valenta, D. Adrian, A. Sanso, S. Cohney, J. Fried, M. Hastings, J. A. Halderman, and N. Heninger, "Measuring small subgroup attacks against diffie-hellman (eprint)," in *Proceedings of 39th IEEE Symposium on Security and Privacy (Oakland'18)*, 2018. (`https://eprint.iacr.org/2016/995.pdf`)

[41] P. Wang, I. Osipkov, and Y. Kim, *Myrmic: Secure and Robust DHT Routing*, 2007. (`https://www.dtc.umn.edu/publications/reports/2006_20.pdf`)

[42] C. C. Yang, T. Y. Chang, and M. S. Hwang, "A (t,n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, Apr. 2004.

Biography

Dr. Daouda Ahmat completed his Computer Science PhD studies in 2014 at the University of Bordeaux, France. He also graduated (MSc) from the same University in 2011 after having obtained a Bachelor degree from the University of NDjamena, Chad. He is currently the Vice-Rector of Virtual University of Chad. His research interests include Network Security, Mobile VPN, Key Exchange in Distributed Systems, Peer-to-Peer Networks,

Anonymous Systems, Blockchain and ICT for Development.

Mr. Marayi Choroma is an Industrial and Computer Engineer. He obtained a University Diploma of Research in Digital Education from the University of Lille 1, France in 2015. He is currently interested in the design and implementation of digital instruments for learning and teaching, including the integration of MOOCs into higher education programs in Chad.

Dr. Tégawendé F. Bissyandé is a research scientist at the Interdisciplinary Centre for Security, Reliability and Trust of the University of Luxembourg. He received his PhD degree in Computer Sciences from the University of Bordeaux in 2013. His main research interests are in software engineering, notably debugging software to repair bugs, patch vulnerabilities and identify security threats such as malware.