

An Authenticated Key Scheme over Elliptic Curves and Security Considerations

Riccardo Aragona · Roberto Civino ·
Norberto Gavioli · Marco Pugliese

Received: date / Accepted: date

Abstract Nodes of sensor networks may be resource-constrained devices, often having a limited lifetime, making sensor networks remarkably dynamic environments. Managing a cryptographic protocol on such setups may require a disproportionate effort when it comes to update the secret parameters of new nodes that enter the network in place of dismantled sensors. For this reason, the designers of schemes for sensor network are always concerned with the need of scalable and adaptable solutions. In this work, we present a novel elliptic-curve based solution, derived from the previously released cryptographic protocol TAKS, which addresses this issue. We give a formal description of the scheme, built on a two-dimensional vector space over a prime field and over elliptic curves, where node topology is more relevant than node identity, allowing a dynamic handling of the network and reducing the cost of network updates. We also study some security concerns and their relation to the related discrete logarithm problem over elliptic curves.

Keywords Key Establishment Protocols · Elliptic Curve Cryptography · Discrete Logarithm Problem · Authenticated Networks · Directed Graphs

Mathematics Subject Classification (2010) 94A60 · 94A62 · 94C15

R. Aragona - ORCID iD: 0000-0001-8834-4358
DISIM - University of L'Aquila, Via Vetoio, 67100, L'Aquila (AQ) - Italy
E-mail: riccardo.aragona@univaq.it

R. Civino - ORCID iD: 0000-0003-3672-8485
DISIM - University of L'Aquila, Via Vetoio, 67100, L'Aquila (AQ) - Italy
E-mail: roberto.civino@univaq.it

N. Gavioli - ORCID iD: 0000-0001-9500-0860
DISIM - University of L'Aquila, Via Vetoio, 67100, L'Aquila (AQ) - Italy
E-mail: norberto.gavioli@univaq.it

M. Pugliese
DISIM - University of L'Aquila, Via Vetoio, 67100, L'Aquila (AQ) - Italy
E-mail: marpug@univaq.it

1 Introduction

The *Topology Authenticated Key Scheme* (TAKS) is a cryptographic protocol, proposed in [22,24] for the first time and successively generalised [17, 18,23], providing security over a resource-constrained network (typically ad-hoc networks, e.g. sensor networks for monitoring services). Its authentication mechanism is based on *node topology* rather than on *node identity*, due to the limited lifetime of nodes in a resource-constrained network. Indeed, while nodes in infra-structured networks can rely on an external power supply and on a stable planned maintenance service with human intervention, the nodes in ad-hoc networks can rely only on their own battery or some other energy harvesting mechanism, and maintenance services are usually remotely performed without human intervention. When an off-duty node is replaced with a new node, the new node identity enters in the network, but node topology remains unchanged and the authentication mechanism does not need any updating. Other examples of network-based key pre-distribution schemes may be found in [1,2,8,14,15]. Vector spaces, rather than scalars, are introduced to allow the setup of a truly scalable point-to-multipoint communication sessions, hence without setting up multiple point-to-point sessions: the added dimensionality introduces a further degree of freedom for the CA in the procedure of parameters computation and assignment. From an engineering point of view, point-to-multipoint sessions provide a relevant feature, especially for clustered networks, where time synchronisation in data transmission is required as well as lighter memory storage: point-to-multipoint sessions avoid traffic flooding, hence wasting of energy, when maintenance services are activated on such clustered networks (e.g. the updating of some configuration parameters in a specific portion of the network).

The scheme we propose in this paper, called *Elliptic Curve based Topology Authenticated Key Scheme* (ECTAKS), is derived from TAKS and is defined as a hybrid deterministic *Key Establishment Protocol* (KEP) over elliptic curves, and is designed to establish both point-to-point and point-to-multipoint secure links among nodes. Security features of ECTAKS may include confidentiality (data encryption), data integrity and sender authentication (signature). Other examples of hybrid KEPs may be found in [9,10,16]. In ECTAKS the shared secret is a symmetric key generated by each party involved in the communication session upon a successful authentication process, where each party verifies if the other party belongs to its authenticated network. Such a network is represented as a graph, where parties (network nodes) are the vertices and the communication links are the edges. The assignment parameter to each node is carried out by an external *Certification Authority* (CA); the scheme parameter are successively preloaded into the nodes.

While TAKS [24] provides only key establishment facilities for point-to-point communications by means of a Diffie-Hellman-like scheme and its generalisation [17] extends to point-to-multipoint sessions, the improvements implemented in this paper directly provide ECTAKS with key establishment capa-

bilities for both point-to-point and point-to-multipoint communications. More importantly, elliptic curve cryptography allows achieving comparable security levels using reduced key lengths [7].

In this paper we provide a rigorous description of ECTAKS and address a security analysis of the scheme. In this regard, we will show that ECTAKS can be broken if an adversary can solve the intractable Discrete Logarithm problem over elliptic curves, provided that it also manage to solve a linear system of equations.

The paper is organised as follows: in Sec. 2 we introduce the notation and some auxiliary results. The scheme is defined in Sec. 3, together with the authenticated-encryption methods which is derived from it. The security analysis of the scheme is carried out in Sec. 4. Sec. 5 concludes the paper with some considerations on open problems.

2 Notation

ECTAKS is a scheme based on elliptic curves as well as vector spaces over finite fields. The networks of nodes where ECTAKS is built is modeled by a graph. In order to provide a rigorous description of the scheme and of its model, let us define the following elements.

Spaces Let p be a prime number, and let \mathbb{F}_p be the finite field with p elements.

We denote by $\alpha \stackrel{\text{R}}{\in} \mathbb{F}_p$ a uniformly random generated element in \mathbb{F}_p . The scheme presented in this paper is mainly based on the 2-dimensional vector space $(\mathbb{F}_p)^2$. Scalar elements in \mathbb{F}_p are usually denoted by lower-case greek letters, whereas vectors in $(\mathbb{F}_p)^2$ are denoted by bold latin letters. Given two vectors $\mathbf{k} = (\alpha_1, \alpha_2)$, $\mathbf{t} = (\beta_1, \beta_2) \in (\mathbb{F}_p)^2$ we define the scalar product over \mathbb{F}_p of \mathbf{k} and \mathbf{t} as

$$\mathbf{k} \cdot \mathbf{t} \stackrel{\text{def}}{=} \alpha_1 \beta_1 + \alpha_2 \beta_2 \in \mathbb{F}_p.$$

Elliptic curves Let $q > 3$ be the power of a prime number. An elliptic curve over \mathbb{F}_q is the abelian group of \mathbb{F}_q -rational points satisfying a Weierstrass equation [25], i.e.

$$\mathcal{C} \stackrel{\text{def}}{=} \{(x, y) \in (\mathbb{F}_q)^2 \mid y^2 = x^3 + ax + b\},$$

where $a, b \in \mathbb{F}_q$. Throughout this paper we will denote by $G \in \mathcal{C}$ the generator of a subgroup of \mathcal{C} of order p , called the *base point*. Given a vector $\mathbf{k} = (\alpha_1, \alpha_2) \in (\mathbb{F}_p)^2$ and $\beta_1, \beta_2 \in \mathbb{F}_p$ we define

$$\mathbf{k}G \stackrel{\text{def}}{=} \underbrace{(G + G + \dots + G)}_{\alpha_1 \text{ times}}, \underbrace{(G + G + \dots + G)}_{\alpha_2 \text{ times}} = (\alpha_1 G, \alpha_2 G) \in \mathcal{C}^2,$$

and

$$\mathbf{k} \cdot (\beta_1 G, \beta_2 G) \stackrel{\text{def}}{=} \alpha_1 \beta_1 G + \alpha_2 \beta_2 G \in \mathcal{C}.$$

The following result is easily checked.

Lemma 1 Let $\mathbf{k}, \mathbf{t} \in (\mathbb{F}_p)^2$. Then we have

$$(\mathbf{k} \cdot \mathbf{t})G = \mathbf{k} \cdot (\mathbf{t}G).$$

Proof Let $\mathbf{k} = (\alpha_1, \alpha_2)$ and $\mathbf{t} = (\beta_1, \beta_2)$. Then

$$\begin{aligned} (\mathbf{k} \cdot \mathbf{t})G &= (\alpha_1\beta_1 + \alpha_2\beta_2)G \\ &= \alpha_1\beta_1G + \alpha_2\beta_2G \\ &= (\alpha_1, \alpha_2) \cdot (\beta_1G, \beta_2G) \\ &= \mathbf{k} \cdot (\mathbf{t}G). \end{aligned}$$

□

We assume the security of the scheme we propose in this paper to be relying on the following security assumption [11,21].

Problem 1 (ECDL problem) Let \mathcal{C} be an elliptic curve over the finite field \mathbb{F}_q , let $P \in \mathcal{C}$ and $m > 1$. The (*computational*) *Elliptic Curve Discrete Logarithm problem* (ECDL problem) is the problem of finding the integer m when the points P and mP are given.

Provided that the curve meets some well-established requirements (see e.g. [4, 5, 12, 20]), the ECDL problem is assumed to be computationally intractable [3, 13]. For some overviews on algorithms solving the ECDL problem and related problems, see e.g. [6, 19].

Graphs A *directed graph* is a pair of sets (V, E) , where $V \neq \emptyset$ and $E \subseteq V \times V$. We call *arrows* the elements of E , and for each arrow $e = (i, j) \in E$ we denote the *tail* of the arrow as $t(e) \stackrel{\text{def}}{=} i$ and the *head* of the arrow as $h(e) \stackrel{\text{def}}{=} j$. In order to keep the notation cleaner, we will sometimes denote the arrow (i, j) by writing “ $i - j$ ”. If $i, j \in V$, we say that i and j are *connected* in (V, E) if $(i, j) \in E$ and $(j, i) \in E$.

3 The scheme

The *Elliptic Curve Topology Authenticated Key Scheme* (ECTAKS) is a cryptographic protocol based on a network of users who want to communicate with each others. The network is modeled by means of a graph where users are represented as nodes. Two users can communicate if and only if they are connected in the corresponding graph. The communication between two users can start if they manage to exchange a shared secret which can be used to instantiate a Diffie-Hellman-like protocol. In this setting, if the arrow from node i to node j exists, then user i is allowed to start a communication session with user j . The external CA assigns a set of parameters, called *Local Configuration Data* (LCD), to each node of the network. For each node, LCD is composed by two secret components, that remain unchanged once generated, and a public component, which is updated every time a new node joins

the dynamic network. With respect to classical solutions of key-assignment schemes on network, the scheme proposed in this paper does not only rely on pre-distribution of keys in nodes. Indeed, the proposed protocol is rather based on a dynamic assignment of the public components to each node and on a static assignment of secret components. The shared secret in ECTAKS is in fact a function of both sender and receiver private key components, while in ephemeral Diffie-Hellman-like schemes the shared secret is usually a function of the complete sender private key and the receiver public key. Moreover, the CA generates LCD in such a way communications among nodes are allowed only if their topology is compliant to the planned network topology.

3.1 Parameter definition

Let N be a positive integer and let $V \stackrel{\text{def}}{=} \{1, 2, \dots, N\}$. The *authenticated network topology* is a symmetric directed graph $\text{ANT} = (V, E)$, i.e. a graph where $E \subset V \times V$ and if $(i, j) \in E$, then $(j, i) \in E$. We furthermore assume that ANT is loop-free, i.e. without cycles of length 1. For each $1 \leq i \leq N$, $\text{ANT}_i = (V_i, E_i)$ is the (non-symmetric and cycle-free) directed subgraph of ANT such that

$$E_i \stackrel{\text{def}}{=} \{e \in E \mid t(e) = i\} \quad \text{and} \quad V_i \stackrel{\text{def}}{=} \{i\} \cup \{h(e) \mid e \in E_i\}.$$

In the point of view of our application, ANT_i is the subgraph of the users which user i is entitled to communicate to. An example of network topology network is depicted in Fig. 1.

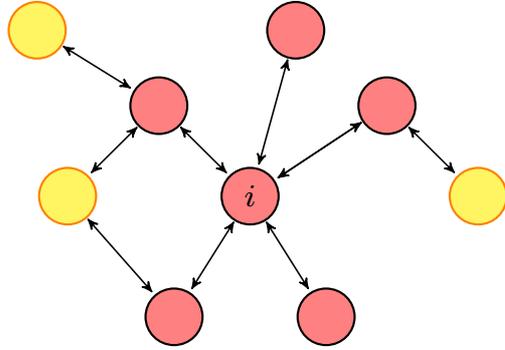


Fig. 1 An example of ANT , where red nodes represent ANT_i .

Let us now denote by \mathcal{C} an elliptic curve over \mathbb{F}_q , where $q > 3$, and $G \in \mathcal{C}$ is the base point, whose order is prime number p . From now on we will assume $p \gg N$. Once the ANT has been established, the CA is in charge of the

assignment of the scheme parameters to each node. For each node $i \in V$, its assigned *local configuration data* $\text{LCD}_i = (S_i, P_i)$ is such that

$$S_i \stackrel{\text{def}}{=} \{\mathbf{k}_i, \mathbf{t}_i\} \quad \text{and} \quad P_i \stackrel{\text{def}}{=} \{\mathbf{m}_{i-j}G\}_{j \in V_i \setminus \{i\}},$$

where $\mathbf{k}_i \in (\mathbb{F}_p)^2 \setminus \{\mathbf{0}\}$ is called the *local key component* corresponding to the node i , $\mathbf{t}_i \in (\mathbb{F}_p)^2 \setminus \{\mathbf{0}\}$ is called the *transmitted key component* corresponding to the node i , and $\mathbf{m}_{i-j}G \in \mathcal{C}^2 \setminus \{0\}$ is called the *topology vector* corresponding to the arrow (i, j) . The component S_i represents the private information assigned to node i , whereas P_i represents its public information.

The CA assigns the parameters to each node in a sequential way, once it has chosen an arbitrary *root* node for each connected component of the graph. Starting from the parameters assigned to the root node, the CA computes the parameters for the other nodes of the graph according to some constraints which allow each pair of topologically admissible nodes to compute a shared secret, that we called *Elliptic Curve Topology Authenticated Key* (ECTAK).

3.2 Parameter assignment and shared secrets

For sake of simplicity the root node where parameter assignment starts is here assumed to be node 1.

Node 1 The parameters \mathbf{k}_1 and \mathbf{t}_1 are generated randomly from $(\mathbb{F}_p)^2 \setminus \{\mathbf{0}\}$ and assigned to the secret component S_1 of node 1.

Node 2 Let us assume that the second node chosen by the CA is node 2 in V_1 . Then, the parameter \mathbf{m}_{1-2} is generated randomly by the CA, provided that $\mathbf{k}_1 \cdot \mathbf{m}_{1-2} \neq 0$, and the corresponding topology vector $\mathbf{m}_{1-2}G$ is appended to the public component P_1 of node 1. Notice that CA can generate a random vector until the previous condition is satisfied. Once the topology vector related to the arrow $(1, 2)$ has been defined, the parameter generation for node 1 is complete and the parameters for node 2 can be defined running the following steps:

- the parameter \mathbf{k}_2 is randomly chosen by the CA from the solutions of the linear equation

$$\mathbf{k}_1 \cdot \mathbf{m}_{1-2} = \mathbf{k}_2 \cdot \mathbf{t}_1 \tag{1}$$

and it is assigned to node 2;

- the parameter \mathbf{m}_{2-1} related to the arrow $(2, 1)$ is generated randomly, provided that

$$\mathbf{k}_2 \cdot \mathbf{m}_{2-1} \neq 0;$$

the corresponding topology vector $\mathbf{m}_{2-1}G$ is assigned to node 2;

- \mathbf{t}_2 is randomly chosen by the CA from the solutions of the linear equation

$$\mathbf{k}_2 \cdot \mathbf{m}_{2-1} = \mathbf{k}_1 \cdot \mathbf{t}_2 \tag{2}$$

and it is assigned to node 2.

At the end of this process the CA has assigned:

- $S_1 = \{\mathbf{k}_1, \mathbf{t}_1\}$, $P_1 = \{\mathbf{m}_{1-2}G\}$ to node 1,
- $S_2 = \{\mathbf{k}_2, \mathbf{t}_2\}$ and $P_2 = \{\mathbf{m}_{2-1}G\}$ to node 2.

Assume now that node 1 wants to start a session with node 2. Then node 1 and node 2 can agree on an ephemeral shared secret, performing the following operations:

- node 1 generates a random non-zero element $\alpha \stackrel{\text{R}}{\in} \mathbb{F}_p$;
- node 1 sends $\alpha \mathbf{t}_1 G$ to node 2;
- node 1 defines $\text{ECTAK}_{1-2} \stackrel{\text{def}}{=} \alpha \mathbf{k}_1 \cdot (\mathbf{m}_{1-2}G)$.

Now node 2 can compute

$$\mathbf{k}_2 \cdot (\alpha \mathbf{t}_1 G) = (\mathbf{k}_2 \cdot \alpha \mathbf{t}_1)G = (\alpha \mathbf{k}_1 \cdot \mathbf{m}_{1-2})G = \alpha \mathbf{k}_1 \cdot (\mathbf{m}_{1-2}G) = \text{ECTAK}_{1-2},$$

where the second equality is obtained from Eq. (1). Consequently node 1 and node 2 have shared the non-zero secret $\text{ECTAK}_{1-2} \in \mathcal{C}$. Similarly, node 2 can agree with node 1 on the shared secret

$$\text{ECTAK}_{2-1} \stackrel{\text{def}}{=} \alpha \mathbf{k}_2 \cdot (\mathbf{m}_{2-1}G) = \mathbf{k}_1 \cdot (\alpha \mathbf{t}_2 G),$$

where α is again an ephemeral random chosen non-zero element in \mathbb{F}_p generated by node 2, and the second equality is derived from Eq. (2).

Node 3 Assume now that node 1 wants to agree on a shared secret with another node in V_1 , e.g. node 3. Then the secret component S_1 of node 1 remains unchanged and, proceeding as for node 2, the parameter \mathbf{m}_{1-3} is generated randomly by the CA, provided that $\mathbf{k}_1 \cdot \mathbf{m}_{1-3} \neq 0$, and the corresponding topology vector $\mathbf{m}_{1-3}G$ is appended to the public component P_1 of node 1. Once the topology vector related to the arrow (1,3) has been defined, the parameters \mathbf{k}_3 , \mathbf{m}_{3-1} and \mathbf{t}_3 for node 3 can be defined proceeding as for node 2. At the end of this process the secret and public components are respectively

- $S_1 = \{\mathbf{k}_1, \mathbf{t}_1\}$ and $P_1 = \{\mathbf{m}_{1-2}G, \mathbf{m}_{1-3}G\}$ for node 1, and
- $S_3 = \{\mathbf{k}_3, \mathbf{t}_3\}$ and $P_3 = \{\mathbf{m}_{3-1}G\}$ for node 3.

Consequently, as in the case of the shared secret between node 1 and node 2, node 1 and node 3 can share the non-zero secret

$$\text{ECTAK}_{1-3} \stackrel{\text{def}}{=} \alpha \mathbf{k}_1 \cdot (\mathbf{m}_{1-3}G) = \mathbf{k}_3 \cdot (\alpha \mathbf{t}_1 G),$$

where α is an ephemeral random chosen non-zero element in \mathbb{F}_p generated by node 1. Similarly, node 3 can agree with node 1 on the shared secret

$$\text{ECTAK}_{3-1} \stackrel{\text{def}}{=} \alpha \mathbf{k}_3 \cdot (\mathbf{m}_{3-1}G) = \mathbf{k}_1 \cdot (\alpha \mathbf{t}_3 G),$$

where α is again an ephemeral random chosen non-zero element in \mathbb{F}_p generated by node 3.

Generic node j In the general case, suppose that the CA has generated $S_i = \{\mathbf{k}_i, \mathbf{t}_i\}$ for node i , as described above, and i wants to agree on a shared secret with node j in V_i . Then, the parameter \mathbf{m}_{i-j} is generated randomly by the CA, provided that $\mathbf{k}_i \cdot \mathbf{m}_{i-j} \neq 0$, and the corresponding topology vector $\mathbf{m}_{i-j}G$ is appended to the public component P_i of node i . Once the topology vector related to the arrow (i, j) has been defined, the parameters for node j can be defined running the following steps:

- the parameter \mathbf{k}_j is randomly chosen by the CA from the solutions of the linear equation

$$\mathbf{k}_i \cdot \mathbf{m}_{i-j} = \mathbf{k}_j \cdot \mathbf{t}_i \quad (3)$$

and it is assigned to node j ;

- the parameter \mathbf{m}_{j-i} related to the arrow (j, i) is generated randomly, provided that

$$\mathbf{k}_j \cdot \mathbf{m}_{j-i} \neq 0;$$

the corresponding topology vector $\mathbf{m}_{j-i}G$ is assigned to node j ;

- \mathbf{t}_j is randomly chosen by the CA from the solutions of the linear equation

$$\mathbf{k}_j \cdot \mathbf{m}_{j-i} = \mathbf{k}_i \cdot \mathbf{t}_j \quad (4)$$

and it is assigned to node j .

At the end of this process we have that

- $S_i = \{\mathbf{k}_i, \mathbf{t}_i\}$ and $\mathbf{m}_{i-j}G$ is appended to P_i for node i , and
- $S_j = \{\mathbf{k}_j, \mathbf{t}_j\}$ and $\mathbf{m}_{j-i}G$ is appended to P_j for node j .

Assume now that node i wants to start a session with node j . Then node i and node j can agree on an ephemeral shared secret, performing the following operations:

- node i generates a random non-zero element $\alpha \in \mathbb{F}_p^{\mathbb{R}}$;
- node i sends $\alpha \mathbf{t}_i G$ to node j ;
- node i defines $\text{ECTAK}_{i-j} \stackrel{\text{def}}{=} \alpha \mathbf{k}_i \cdot (\mathbf{m}_{i-j}G)$.

Now node j can compute

$$\mathbf{k}_j \cdot (\alpha \mathbf{t}_i G) = (\mathbf{k}_j \cdot \alpha \mathbf{t}_i)G = (\alpha \mathbf{k}_i \cdot \mathbf{m}_{i-j})G = \alpha \mathbf{k}_i \cdot (\mathbf{m}_{i-j}G) = \text{ECTAK}_{i-j},$$

where the second equality is obtained from Eq. (3). Consequently node i and node j have shared the non-zero secret $\text{ECTAK}_{i-j} \in \mathcal{C}$. Similarly, node j can agree with node i on the shared secret

$$\text{ECTAK}_{j-i} \stackrel{\text{def}}{=} \alpha \mathbf{k}_j \cdot (\mathbf{m}_{j-i}G) = \mathbf{k}_i \cdot (\alpha \mathbf{t}_j G),$$

where α is again an ephemeral random chosen non-zero element in \mathbb{F}_p generated by node j , and the second equality is derived from Eq. (4).

Remark 1 For each $(i, j) \in E$, the component \mathbf{m}_{i-j} , which is generated by the CA in order to define the public component $\mathbf{m}_{i-j}G$, is not accessible by any user (belonging or not to the network), unless they can solve the ECDL problem, as better explained in Section 4.

Remark 2 When the session between node i and node j has timed out or is not anymore valid, node i and node j can again agree on a disposable shared secret by selecting a new random parameter α . The same happens if node i is damaged and needs to be replaced by another sensor. The CA assigns to the new node i the same secret parameter of the former node i , and the communication with node j is again established by selecting a new random parameter α .

Remark 3 The parameter assignment is highly scalar-product based. For this reason, it is important to point out that for each $i, j \in V$, the products of Eq. (3) defining ECTAK_{i-j} are uniformly distributed over $\mathbb{F}_p \setminus \{0\}$, since, by definition, its inputs are non-zero elements of $(\mathbb{F}_p)^2$. Moreover, the secret components \mathbf{k}_i and \mathbf{t}_i can be chosen from the p solutions of Eq. (3) and Eq. (4) respectively. In other words, the constraints defined in Eq. (3) and Eq. (4) reduce the complexity of guessing the secret component \mathbf{k}_i (respectively \mathbf{t}_i) from p^2 to p , since the value to be guessed needs to satisfy a linear equation. However, this does not represent a security issue since the parameter p is chosen to be a secure parameters, therefore the security of the scheme should rely on the size of p and not on the size of p^2 .

3.3 ECTAK-based authenticated encryption

We show here a classical way to provide authenticated encryption, using the ECTAK as shared secret. In the following we denote by H and KDF a keyed hash function and a key-derivation function respectively, whereas Enc_k and Dec_k respectively denotes the encryption and the decryption procedures using the key k of a symmetric encryption method, where $\text{Dec}_k = \text{Enc}_k^{-1}$.

Assume now that node 1 wants to send a signed encrypted message m to node 2. Then node 1 performs the following operations:

- generates $\alpha \stackrel{\text{R}}{\in} \mathbb{F}_p \setminus \{0\}$;
- computes $\text{ECTAK}_{1-2} \stackrel{\text{def}}{=} \alpha \mathbf{k}_1 \cdot (\mathbf{m}_{1-2}G)$;
- computes $(k_1, k_2) \stackrel{\text{def}}{=} \text{KDF}(\text{ECTAK}_{1-2}, \alpha \mathbf{t}_1G)$;
- computes $c \stackrel{\text{def}}{=} \text{Enc}_{k_1}(m)$;
- computes $s \stackrel{\text{def}}{=} H(k_2, c)$;
- sends $(\alpha \mathbf{t}_1G, c, s)$ to node 2,

where the size of m, c and s suits respectively the domain of encryption and hash functions.

Node 2 can perform the following steps:

- computes $\mathbf{k}_2 \cdot (\alpha \mathbf{t}_1G) = \text{ECTAK}_{1-2}$;
- recovers (k_1, k_2) by computing $\text{KDF}(\text{ECTAK}_{1-2}, \alpha \mathbf{t}_1G)$;
- recovers m computing $\text{Dec}_{k_1}(c)$;
- checks that $s = H(k_2, c)$.

4 Considerations on security

In this section we present some security properties of the scheme. We will show, in Theorem 1, in which way an attacker can successfully determine the secret parameters of a target node i . We will prove that, in order the attack to be successful, the attacker needs to recover the secret information of at least two nodes connected to node i . The success probability of the attack is calculated in Theorem 2. The attack relies on the ability of the attacker to solve an instance of the intractable ECDL problem. To the best of our current state of knowledge, it is not possible to provide a formal reduction from one problem to the other.

Let us now prove our result showing that if an attacker can solve the ECDL problem and can recover the secret components S_j and S_k , for some nodes $j, k \in V_i$ connected to i , then it can recover S_i only if an algebraic condition on the coordinates is satisfied. In particular, from this follows that compromising one and only node is not enough to recover S_i . Without any loss of generality, let us denote by node 1, node 2 and node 3 the three nodes previously mentioned. Let us assume that node 1 is targeted by the attacker, which has successfully recovered data from node 2 and node 3. Moreover, to further simplify, let us assume $\text{ANT}_1 = (V_1, E_1)$, where $V_1 = \{1, 2, 3\}$ and $E_1 = \{(1, 2), (1, 3)\}$, as depicted in Fig. 2.

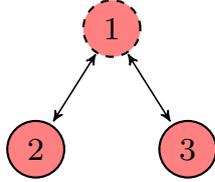


Fig. 2 The network targeted by the attacker, where the target node 1 is highlighted.

We assume that $\text{LCD}_2 = \{(\mathbf{k}_2, \mathbf{t}_2), \mathbf{m}_{2-1}G\}$ and $\text{LCD}_3 = \{(\mathbf{k}_3, \mathbf{t}_3), \mathbf{m}_{3-1}G\}$ are known to the attacker aiming at recovering $\mathbf{k}_1, \mathbf{t}_1 \in \text{LCD}_1$, recalling that $\mathbf{m}_{1-2}G, \mathbf{m}_{1-3}G$ are publicly available.

$$\begin{aligned} (\mathbf{k}_2 \cdot \mathbf{t}_1)G &= \mathbf{k}_1 \cdot (\mathbf{m}_{1-2}G) \\ (\mathbf{k}_1 \cdot \mathbf{t}_2)G &= \mathbf{k}_2 \cdot (\mathbf{m}_{2-1}G) \\ (\mathbf{k}_3 \cdot \mathbf{t}_1)G &= \mathbf{k}_1 \cdot (\mathbf{m}_{1-3}G) \\ (\mathbf{k}_1 \cdot \mathbf{t}_3)G &= \mathbf{k}_3 \cdot (\mathbf{m}_{3-1}G). \end{aligned}$$

Since the attacker has access to an algorithm which solves the ECDL problem, it can access $\mathbf{m}_{1-2} = (a_{13}, a_{14})$ and $\mathbf{m}_{1-3} = (a_{33}, a_{34})$. Therefore, denoting by $\mathbf{k}_2 = (a_{11}, a_{12})$, $\mathbf{t}_2 = (a_{23}, a_{24})$, $\mathbf{k}_3 = (a_{31}, a_{32})$ and $\mathbf{t}_3 = (a_{43}, a_{44})$, the

previous equations correspond to the following system in the unknowns $\mathbf{t}_1 = (x_1, x_2)$ and $\mathbf{k}_1 = (x_3, x_4)$:

$$\begin{cases} (a_{11}x_1 + a_{12}x_2 - a_{13}x_3 - a_{14}x_4)G = 0 \\ (a_{23}x_3 + a_{24}x_4)G = b_2G \\ (a_{31}x_1 + a_{32}x_2 - a_{33}x_3 - a_{34}x_4)G = 0 \\ (a_{43}x_3 + a_{44}x_4)G = b_4G, \end{cases} \quad (5)$$

where $b_2 = \mathbf{k}_2 \cdot \mathbf{m}_{2-1}$ and $b_4 = \mathbf{k}_3 \cdot \mathbf{m}_{3-1}$. Therefore, denoting by

$$A \stackrel{\text{def}}{=} \begin{pmatrix} a_{11} & a_{12} & -a_{13} & -a_{14} \\ 0 & 0 & a_{23} & a_{24} \\ a_{31} & a_{32} & -a_{33} & -a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{pmatrix}, \quad (6)$$

$b \stackrel{\text{def}}{=} (0, b_2, 0, b_4)^t$ and $x \stackrel{\text{def}}{=} (x_1, x_2, x_3, x_4)^t$, the system in Eq.(5) is equivalent to the linear equation

$$Ax = b. \quad (7)$$

Theorem 1 *Let Adv be an adversary then can solve the ECDL problem. If $\det(A) \neq 0$, then Adv can recover S_1 .*

Proof Since the adversary can solve the ECDL problem, it can build the system $Ax = b$ of Eq.(7). The result trivially follows, since $\det(A) \neq 0$ implies that the system admits one and only one solution. \square

Remark 4 Notice that, due to the requirements on the LCD on each node, the first and the second row of A are linearly independent, and the same holds for the third and the fourth. If Adv can access the secret components of two nodes, then, since $\text{rk}(A|b) = \text{rk}(A) \in \{2, 4\}$, the system in Eq.(5) as at least one solution. Moreover, if $\text{rk} A = 2$, the system admits p^2 solutions, then the method of Theorem 1 leads to an attack to the scheme with success probability $1/p^2$. Indeed, if the attacker selects one of the $p^2 - 1$ solutions of the system which do not match the correct secret component of node 1, then the attempted impersonation attack is easily disclosed in the authenticated-encryption phase of the protocol (see Sec. 3.3). The same holds if only one node is compromised by Adv, since only two equations of the system are known.

We will now show that the success probability of the attack described in Theorem 1 approaches 1 when the prime p is sufficiently large.

Theorem 2 *Let \mathcal{S}_p be the probability that Adv successfully recovers the secret component of S_1 using the method of Theorem 1. Then*

$$\lim_{p \rightarrow \infty} \mathcal{S}_p = 1.$$

Proof Let us recall that, for each parameter assignment in a scheme with 3 nodes, we can construct a matrix as in Eq. (6). We call such a matrix an *admissible* matrix for the scheme. Let us denote by \mathcal{A}_p be the set of matrices in $(\mathbb{F}_p)^{4 \times 4}$ which are admissible, and by \mathcal{I}_p the subset of those which are invertible. Then we have

$$S_p = \frac{\#\mathcal{I}_p}{\#\mathcal{A}_p} > \frac{\#\mathcal{I}_p}{p^{12}}. \quad (8)$$

Let us now count $\#\mathcal{I}_p$. The parameters \mathbf{k}_1 and \mathbf{t}_1 are chosen randomly in $(\mathbb{F}_p)^2 \setminus \{\mathbf{0}\}$, whereas $\mathbf{m}_{1-2} \in (\mathbb{F}_p)^2 \setminus \{\mathbf{0}\}$ is chosen such that $\mathbf{k}_1 \cdot \mathbf{m}_{1-2} \neq 0$. The coefficient \mathbf{k}_2 is chosen such that $\mathbf{k}_2 \cdot \mathbf{t}_1 \neq 0$, since $\mathbf{k}_2 \cdot \mathbf{t}_1 = \mathbf{k}_1 \cdot \mathbf{m}_{1-2}$. Notice that, if $\mathbf{t}_1 = (x_1, x_2)$, then we need to rule out from the possible choices of $\mathbf{k}_2 = (a_{11}, a_{12})$ those which satisfy $a_{11}x_1 = -a_{12}x_2$. This reduces to $p^2 - p$ the possibilities for the vector \mathbf{k}_2 . Now, \mathbf{m}_{2-1} can be chosen making sure that $\mathbf{k}_2 \cdot \mathbf{m}_{2-1} \neq 0$. Since \mathbf{k}_2 is not fixed, we have $p^2 - 1$ possible choices for \mathbf{m}_{1-2} . Analogously, \mathbf{t}_2 can be chosen in $p^2 - p$ ways, since the value $\mathbf{k}_1 = (x_3, x_4)$ is fixed and $\mathbf{k}_1 \cdot \mathbf{t}_2 \neq 0$. Hence, if the solution $(x_1, x_2, x_3, x_4) \in (\mathbb{F}_p)^4$ is fixed, we obtain $\mathbf{k}_2, \mathbf{t}_2, \mathbf{m}_{2-1}$ in $(p^2 - p)^2(p^2 - 1)$ way. The same holds when considering $\mathbf{k}_3, \mathbf{t}_3, \mathbf{m}_{3-1}$. Noticing that in this argument we are using the fact that the constructed matrix is invertible, since we are assuming that (x_1, x_2, x_3, x_4) is the unique solution of the problem, we obtain

$$\mathcal{I}_p = (p^2 - p)^4(p^2 - 1)^2 = (\#\text{GL}(2, p)^2) (p^2 - p)^2.$$

The result follows from Eq. (8), considering the limit for $p \rightarrow \infty$. \square

Notice that ECTAKS can be defined over an \mathbb{F}_p -vector space of dimension $d > 2$ similarly to the way it was built in Section 3 for an \mathbb{F}_p -vector space of dimension 2, and Theorems 1 and 2 can be extended to the d -dimensional case as well. In particular, it is possible to construct a $2d \times 2d$ triangular block matrix A that, for a large p , is invertible with probability close to 1. Moreover, an attacker who can solve the ECDL problem can successfully determine the secret parameters of a target node i , provided that it can recover the secret components of at least d nodes connected to node i and $\det(A) \neq 0$.

5 Conclusion and future works

In this paper we have introduced the protocol ECTAKS, derived from [24] and here adapted to the case of elliptic-curve cryptography. We have studied some security issues of the scheme, with a focus on the underlying ECDL problem. We have proven that, even though the secret and public components of the scheme are linked by means of linear equations, an attacker who wants to make use of the linear algebra method (explained in Sec. 4) to recover the secret components to a target node needs to be able to solve the ECDL problem and to access the secret components of at least two nodes connected to the target node. Although at the time of writing we understand that the scheme lacks

of a general and complete security proof, the search for an argument showing that an attack to ECTAKS can be converted into an attack to the underlying ECDL problem remains open.

References

1. M. Al-Subaie and M. Zulkernine. Efficacy of Hidden Markov Models Over Neural Networks in Anomaly Intrusion Detection. In *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, volume 1, pages 325–332, 2006.
2. H. Arjmandi and F. Lahouti. A key pre-distribution scheme based on multiple block codes for wireless sensor networks. In *7th International Symposium on Telecommunications (IST'2014)*, pages 857–860, 2014.
3. E. Bach. Intractable problems in number theory. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO' 88*, pages 77–93, New York, NY, 1990. Springer New York.
4. D. J. Bernstein. Curve25519: New Diffie-Hellman Speed Records. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography – PKC 2006*, pages 207–228, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
5. -. FIPS PUB. Digital Signature Standard (DSS). *NIST*, 2013.
6. S. D. Galbraith and P. Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, 78(1):51–72, 2016.
7. V. Gupta, S. Gupta, S. Chang, and D. Stebila. Performance analysis of elliptic curve cryptography for ssl. In *Proceedings of the 1st ACM workshop on Wireless security*, pages 87–94, 2002.
8. Haowen Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *2003 Symposium on Security and Privacy, 2003.*, pages 197–213, 2003.
9. R. J. Kavitha and B. E. Caroline. Hybrid cryptographic technique for heterogeneous wireless sensor networks. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 1016–1020, 2015.
10. D. Kim, J. Yun, and S. Kim. Hybrid public key authentication for wireless sensor networks. In *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 142–143, 2017.
11. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.
12. N. Koblitz. CM-Curves with Good Cryptographic Properties. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, pages 279–287, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.
13. N. Koblitz and A. Menezes. Intractable problems in cryptography. In *Proceedings of the 9th Conference on Finite Fields and Their Applications. Contemporary Mathematics*, volume 518, pages 279–300, 2010.
14. R. Kuchipudi, A. A. M. Qyser, and V. V. S. S. S. Balaram. A dynamic key distribution in wireless sensor networks with reduced communication overhead. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 3651–3654, 2016.
15. D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 03*, pages 52–61, New York, NY, USA, 2003. Association for Computing Machinery.
16. Manjunath CR, S. Anand, and G. Nagaraja. An hybrid secure scheme for secure transmission in grid based Wireless Sensor Network. In *2015 IEEE International Advance Computing Conference (IACC)*, pages 472–475, 2015.
17. S. Marchesani, L. Pomante, M. Pugliese, and F. Santucci. Definition and Development of a Topology-Based Cryptographic Scheme for Wireless Sensor Networks. In M. Zuniga and G. Dini, editors, *Sensor Systems and Software*, pages 47–64, Cham, 2013. Springer International Publishing.
18. S. Marchesani, L. Pomante, F. Santucci, and M. Pugliese. A Cryptographic Scheme for Real-World Wireless Sensor Networks Applications. In *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems, ICCPS 13*, page 249, New York, NY, USA, 2013. Association for Computing Machinery.

19. A. Menezes. The Elliptic Curve Discrete Logarithm Problem: State of the Art. In K. Matsuura and E. Fujisaki, editors, *Advances in Information and Computer Security*, pages 218–218, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
20. A. Menezes, E. Teske, and A. Weng. Weak Fields for ECC. In T. Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, pages 366–386, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
21. V. S. Miller. Use of Elliptic Curves in Cryptography. In H. C. Williams, editor, *Advances in Cryptology – CRYPTO '85 Proceedings*, pages 417–426, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.
22. M. Pugliese. *Managing Security Issues in Advanced Applications of Wireless Sensor Networks*. PhD thesis, Department of Electrical Engineering and Computer Science, University of L'Aquila, 2008. https://mpugliese.webnode.it/_files/200000061-a7608a760b/24.%20phd_thesis.pdf.
23. M. Pugliese, L. Pomante, and F. Santucci. Secure platform over wireless sensor networks. In *Applied Cryptography and Network Security*. IntechOpen, 2012.
24. M. Pugliese and F. Santucci. Pair-wise network topology authenticated hybrid cryptographic keys for Wireless Sensor Networks using vector algebra. In *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 853–859, 2008.
25. J. H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer-Verlag, New York, 2009.