TU Delft

Delft University of Technology

Cryptographic switching functions for multiplicative watermarking in cyber-physical systems

Gallo, Alexander J.; Ferrari, Riccardo M.G.

DOI
10.1016/j.ifacol.2022.07.164

Publication date
2022

Document Version
Final published version

Published in
IFAC-PapersOnline

Citation (APA)
Gallo, A. J., & Ferrari, R. M. G. (2022). Cryptographic switching functions for multiplicative watermarking in cyber-physical systems. *IFAC-PapersOnline*, *55*(6), 414-419. https://doi.org/10.1016/j.ifacol.2022.07.164

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-6 (2022) 414–419

Cryptographic switching functions for multiplicative watermarking in cyber-physical systems ⋆

Alexander J. Gallo◇, and Riccardo M. G. Ferrari◇

◇ *Delft Center for Systems and Control, Mechanical, Maritime, and Materials Engineering, TU Delft, Delft, Netherlands*

(email: a.j.gallo,r.ferrari@tudelft.nl)

Abstract: In this paper we present a novel switching function for multiplicative watermarking systems. The switching function is based on the algebraic structure of elliptic curves over finite fields. The resulting function allows for both watermarking generator and remover to define appropriate system parameters, sharing only limited information, namely a private key. We prove that the resulting watermarking parameters lead to a stable watermarking scheme.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

1. INTRODUCTION

Cyber-physical systems (CPS) are a class of systems characterized by high penetration of computational and communication resources, providing increased performance and efficiency. Recent events have highlighted that the security of these systems, which aptly describe power grids, transport networks, and many other industrial plants, is of critical importance (Falliere et al., 2011; Lee, 2008).

Over the past decade, the control systems community developed a growing body of research to detect, isolate, and mitigate attacks on CPSs (Sandberg et al., 2015; Chen et al., 2020). According to Weerakkody et al. (2019), techniques for cyber-attack detection can be classified as between *passive* or *active* ones: while passive detection methods rely on the effect of the attack and on knowledge of the nominal behavior of the system, in active methods specific signals are constructed to enhance detection capabilities (Weerakkody and Sinopoli, 2015; Ferrari and Teixeira, 2020; Griffioen et al., 2020). Here, we focus on developing a design procedure for an active method: switching multiplicative watermarking (Ferrari and Teixeira, 2020).

In multiplicative watermarking, signals are modulated through a *watermark generator* before being transmitted over the communication network. After transmission, the original signal is reconstructed via a *watermark remover*, before being used for control and diagnostic purposes: this guarantees nominal performances are unchanged. A man-in-the-middle attack affecting the signal during transmission, instead, would cause a reconstruction error, thus leading to detection. Introduced in Ferrari and Teixeira (2020), *switching* multiplicative watermarking is based on the parameters of the watermark generator and remover changing synchronously. This has been proved to make also otherwise stealthy attacks detectable.

In this paper, we aim to design a safe switching function, requiring minimal *secret* information to be shared between the watermark generator and remover, thus enhancing its overall security. We build a switching function based on modular arithmetic of elliptic curves, used in public key exchange cryptography because of the difficulty of solving the discrete logarithm problem. Our contributions are:

a. a switching function for multiplicative watermarking based on elliptic curves, with provable security;
b. guarantees of stability for the resulting watermark generator-remover pair.

The remainder of the paper is structured as follows: in Section 2 we formally introduce the problem formulation; in Section 3 we give an overview of the mathematical properties of modular arithmetic over elliptic curves; in Section 4 we present the algorithm defining the switching function, and analyze its properties. Finally, in Section 5 we provide a numerical implementation of the algorithm. For reasons of space proofs of theoretical results are omitted, but can be found in Gallo and Ferrari (2022).

2. BACKGROUND AND PROBLEM FORMULATION

In this paper we consider a linear time-invariant (LTI) CPS, represented in Figure 1. The closed loop system is composed of a plant $\mathcal{P}$ and a controller $\mathcal{C}$. We consider that the output of the plant, y_p, is transmitted over some form of communication network to $\mathcal{C}$, and is therefore exposed to malicious tampering. Specifically, we consider a man-in-the-middle attack, capable of eavesdropping the communicated signal, as well as injecting false data into the channel. To detect the presence of this attack, we suppose the system is equipped with an anomaly detector $\mathcal{D}$, as well as a watermark generator and remover pair, $\mathcal{W}$ and $\mathcal{Q}$, to enhance the detection properties. Specifically, the measurement output y_p is modulated via an LTI system, and the resulting output, y_w, is transmitted over the communication network. Once received, the signal $\tilde{y}_w$is demodulated via the watermark remover system $\mathcal{Q}$, and the resulting output y_q is used as an input to the controller.

Without loss of generality, we will assume that attacks are present on the sensor side only. Results can be extended easily to the actuator side case.

⋆ This work has been partially supported by the Research Council of Norway through the project AIMWind.

2405-8963 Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2022.07.164

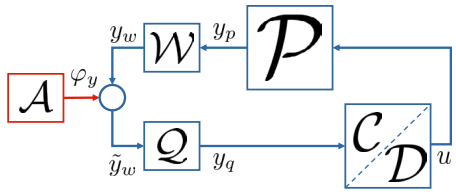

Fig. 1. Block diagram of the closed loop control system including the controller and watermarking systems, and attack on the output communication channel.

2.1 Cyber-physical system

We consider the dynamics of the plant $\mathcal{P}$:

$$\mathcal{P}: \begin{cases} x_p(k+1) = A_p x_p(k) + B_p u(k) + w(k) \\ y_p(k) = C_p x_p(k) + v(k) \end{cases} \tag{1}$$

where $x_p \in \mathbb{R}^{n_x}$ is the state of the plant, $u \in \mathbb{R}^{n_u}$ is the control input, and $y_p \in \mathbb{R}^{n_y}$ is the measurement output. The vectors $w \in \mathbb{R}^{n_x}$ and $v \in \mathbb{R}^{n_y}$ represent the unmodeled process and measurement disturbances. We suppose that all matrices are of the appropriate dimensions. The plant is regulated via a dynamic controller of the form:

$$\mathcal{C}: \begin{cases} x_c(k+1) = A_c x_c(k) + B_c y_q(k) \\ u(k) = C_c x_c(k) + D_c y_q(k) \end{cases} \tag{2}$$

where $x_c \in \mathbb{R}^{n_c}$ is the controller state, and $y_q \in \mathbb{R}^{n_y}$ is the output of the watermarking remover system $\mathcal{Q}$. As mentioned previously, $\mathcal{Q}$ is included in the closed-loop CPS, together with the watermarking generator $\mathcal{W}$, to enhance the detection capabilities of the anomaly detector $\mathcal{D}$, to be introduced. The watermarking pair is defined by:

$$\mathcal{W}: \begin{cases} x_w(k+1) = A_w(\theta_w(k))x_w(k) + B_w(\theta_w(k))y_p(k) \\ y_w(k) = C_w(\theta_w(k))x_w(k) + D_w(\theta_w(k))y_p(k) \end{cases} \tag{3a}$$

$$\mathcal{Q}: \begin{cases} x_q(k+1) = A_q(\theta_q(k))x_q(k) + B_q(\theta_q(k))\tilde{y}_w(k) \\ y_q(k) = C_q(\theta_q(k))x_q(k) + D_q(\theta_q(k))\tilde{y}_w(k) \end{cases} \tag{3b}$$

where the vectors $x_w, x_q \in \mathcal{R}^{n_w}$ and $y_w, y_q \in \mathcal{R}^{n_y}$ are, respectively, the state and outputs of $\mathcal{W}$ and $\mathcal{Q}$.

Finally, the CPS is equipped with an anomaly detection module $\mathcal{D}$, to detect the presence of anomalies (such as malicious false-data injection attacks):

$$\mathcal{D}: \begin{cases} x_r(k+1) = A_r x_r(k) + B_r u(k) + K_r y_q(k) \\ y_r(k) = C_r x_r(k) + L_r y_q(k) \end{cases} \tag{4}$$

where $x_r \in \mathbb{R}^{n_r}$ is the anomaly detector's internal state, and $y_r \in \mathcal{R}^{n_y}$ its output, a residual signal. The test $|y_r(k)| \leq \bar{y}_r(k)$ is computed at each time instant to detect whether an attack is active on the communication network. $\bar{y}_r(k)$ is an appropriately defined, time-varying detection threshold, and the inequality is performed component-by-component. The definition of $\bar{y}_p$ is out of the scope of this paper, but may be found in, e.g., Zhang et al. (2002).

2.2 Attack model

As mentioned previously, we consider y_w to be transmitted over a communication network, and attacked from $k_a > 0$. We define the received signal, appearing in (3b), as:

$$\tilde{y}_w(k) = y_w(k) + \beta_a(k - k_a)\varphi_y(Y_{w,[k-N_a:k]}, k), \tag{5}$$

where $\varphi_y(\cdot)$ is the attack signal, depending on:

$$Y_{w,[k-N_a:k]} \doteq [y_w(k-N_a),\ \ldots,\ y_w(k)]\,; \tag{6}$$

$\beta_a(\cdot)$ is an activation function, defined as a step function.

2.3 Multiplicative watermarking: some background

Let us now focus on the design of the time-varying systems $\mathcal{W}$ and $\mathcal{Q}$, i.e., the watermarking pair. The results reported in this section rely on those in Ferrari and Teixeira (2020).

***Definition** 1.* Two systems $\mathcal{W}$ and $\mathcal{Q}$ (3) are an appropriate watermarking pair if: $\mathcal{W}$ is stable and invertible; $\mathcal{Q}$ is stable; $\theta_w = \theta_q$ implies $\mathcal{Q}$ is the inverse of $\mathcal{W}$. ◁

The definition of the systems $\mathcal{W}$ and $\mathcal{Q}$ in (3) are parametrized by the vectors $\theta_w, \theta_q \in \mathbb{R}^{n_\theta}$. These parameters are piecewise constant, and are updated only at specific switching timeswith the updates given by:

$$\mathcal{W}: \begin{cases} \theta_w^+(k) = \sigma_w(\mathcal{I}_w(k)), & \text{if } \tau_w(y_p(k)) = 1 \\ x_w^+(k) = \rho_w(x_w^-(k), y_p(k), \theta_w^-(k), \theta_w^+(k)) \end{cases} \tag{7a}$$

$$\mathcal{Q}: \begin{cases} \theta_q^+(k) = \sigma_q(\mathcal{I}_q(k)), & \text{if } \tau_q(\tilde{y}_w(k)) = 1 \\ x_q^+(k) = \rho_q(x_q^-(k), \tilde{y}_w^+(k), \theta_q^-(k), \theta_q^+(k)) \end{cases} \tag{7b}$$

where $\mathcal{I}_w(k), \mathcal{I}_q(k)$ are the sets of information available to $\mathcal{W}$ and $\mathcal{Q}$ at time k, as defined in Definitions 3 and 4; for $i \in \{w, q\}$, $\rho_i : \mathbb{R}^{n_w} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_w}$ is a *jump map*, $\sigma_i : \mathbb{R}^{n_y} \to \mathbb{R}^{n_\theta}$ is a *switching map*, and the superscripts $+$ and $-$ denote the values of the vectors before and after a switch. Finally, τ_i are *triggering functions* inducing the switch.

***Definition** 2.* The functions $\tau_q, \tau_w \in \mathbb{R}^{n_y} \to \{0, 1\}$ are said to be trigger functions of $\mathcal{W}$ and $\mathcal{Q}$ if the *triggering sets* $\mathcal{C}_w \doteq \{y_p : \tau_w(y_p) = 1\}$ and $\mathcal{C}_q \doteq \{\tilde{y}_w : \tau_q(\tilde{y}_w) = 1\}$ are convex and open. The sequences $\mathcal{K}_w \doteq \{k : \tau_w(y_p(k)) = 1\}$ and $\mathcal{K}_q \doteq \{k : \tau_q(\tilde{y}_w(k)) = 1\}$ are called the *switching time sequences* of $\mathcal{W}$ and $\mathcal{Q}$. ◁

Given that $\mathcal{W}$ and $\mathcal{Q}$ are not colocated, it is important to properly define what information is available to $\mathcal{W}$ and $\mathcal{Q}$. We therefore formally introduce the *information sets*, $\mathcal{I}_w$ and $\mathcal{I}_q$, defined in terms of input and output data over a window of size $N_I \geq 1, N_I \in \mathbb{Z}_+$, as well as $\theta_i, i \in \{w, q\}$.

***Definition** 3.* (Information at $\mathcal{W}$). The information available to the watermark generator $\mathcal{W}$ is:

$$\mathcal{I}_w(k) = \{Y_{p,k-N_I,k}, x_w(k - N_I), \theta_w(k)\}. \tag{8}$$

◁

***Definition** 4.* (Information at $\mathcal{Q}$). The information available to the watermark remover $\mathcal{Q}$ is:

$$\mathcal{I}_q(k) = \{\tilde{Y}_{w,k-N_I,k}, x_q(k - N_I), \theta_q(k)\}; \tag{9}$$

in addition, for $\kappa_w \in \mathcal{K}_w$, $\mathcal{I}_q^+(\kappa_w) = I_q(\kappa_w) \cup \{y_w^+(\kappa_w)\}$. ◁

***Remark** 1.* Given the definition of the information sets, $\mathcal{I}_q \subset \mathcal{I}_w$ holds, in nominal conditions, for all k. ◁

2.4 Synchronized switching

To guarantee that the watermarking does not influence the closed-loop performance, the following must hold: a. $\mathcal{K}_w = \mathcal{K}_q$ (*synchronized switching times*); b. the output of $\sigma_w(\kappa) = \sigma_q(\kappa)$ and $\rho_w(\kappa) = \rho_q(\kappa)$, for all $\kappa \in \mathcal{K}_w$ (*synchronized switches and jumps*); c. $y_q^+(\kappa) = y_p(\kappa)$ (*synchronized*

output). Here we have slightly abused notation, writing $\sigma_w(k) = \sigma_q(k)$, rather than $\sigma_w(\mathcal{I}_w(k)) = \sigma_q(\mathcal{I}_q(k))$. We consider the same *induced synchronization* scheme presented in Ferrari and Teixeira (2020), where $\mathcal{W}$ triggers a switch and a parameter update in $\mathcal{Q}$ is induced.

***Definition** 5.* (Synchronized watermarking). The watermarking generator $\mathcal{W}$ and remover $\mathcal{Q}$ are synchronized if at switching time k they are: *trigger synchronized*, i.e. $\tau_w(y_p(k)) = \tau_q(\tilde{y}_w(k)) = 1$; *switch synchronized*, i.e. $\theta_w^+(k) = \theta_q^+(k)$; *jump synchronized*, i.e. $x_w^+(k) = x_q^+(k)$; *output synchronized*, i.e. $y_q^+(k) = y_p(k)$. ◁

***Remark** 2.* Under synchronized watermarking $\mathcal{K}_w = \mathcal{K}_q$ holds, and therefore $\mathcal{I}_q^+(\kappa_w) \subset \mathcal{I}_w(\kappa_w), \forall \kappa_w \in \mathcal{K}_w$. ◁

2.5 Problem formulation

Detailed techniques to define $\rho_i(\cdot), i \in \{w, q\}$, as well as design of a minimally *visible* $y_w^+(\kappa_w), \kappa_w \in \mathcal{K}_w$ are presented in Ferrari and Teixeira (2020). However, $\sigma_i(\cdot), i \in \{w, q\}$ are left unspecified. Thus, the objective of this paper is:

***Problem** 1.* Given a switching multiplicative watermarking scheme defined by (3)-(7), define switching functions $\sigma_w(\mathcal{I}_w(k))$ and $\sigma_q(\mathcal{I}_q(k))$ such that: the entire sequence must not be known a priori; $\theta_w^+(k) = \theta_q^+(k)$, for all $k \in \mathcal{K}$; $\theta_i^+(k)$ is such that $\mathcal{W}$ and $\mathcal{Q}$ satisfy Definition 1. △

3. ELLIPTIC CURVES ON FINITE FIELDS: SOME BACKGROUND

Let us now present some background on the arithmetic of elliptic curves. We here rely on overviews presented in Wohlwend (2016); López and Dahab (2000). Elliptic curves are abelian varieties, which have had large success in the field of cryptography: indeed, elliptic curve cryptography (ECC) is a form of *asymmetric* or *public key cryptography*. We exploit the mathematical properties of elliptic curves to define a switching function common to $\mathcal{W}$ and $\mathcal{Q}$ which, although through a shared secret, does not require the switching sequence to be defined *a priori*.

3.1 Some fundamentals in group theory

The following definitions are given to properly introduce the group defined by elliptic curves, and its operations.

***Definition** 6.* A *group* is defined as a set G and a binary operation $\circ$ closed in G, i.e., $a \circ b \in G, \forall a, b \in G$, such that the following axioms (the so called *group axioms*) hold: the operation $\circ$ is associative, i.e., $(a \circ b) \circ c = a \circ (b \circ c)$; an identity element in G, e, exists such that $a \circ e = e \circ a = a$; an inverse element of $a \in G$, denoted a^{-1}, exists under the group operation, such that $a \circ b = b \circ a = e$. ◁

***Definition** 7.* A group G is *abelian* if its binary operator $\circ$ is commutative, i.e. $a \circ b = b \circ a, \forall a, b \in G$. ◁

***Definition** 8.* A group G is said to be *cyclic* if, for an element $h \in G$, every element $g \in G$ satisfies $g = xh$ or $g = h^x$, $x \in \mathbb{Z}_+$, depending on whether the group operation is additive or multiplicative. The element h is said to be the *generator* of the group. ◁

***Definition** 9.* A set $H \subseteq G$ is said to be a cyclic subgroup if it is cyclic, given some generator h. ◁

***Definition** 10.* Suppose $a \in G$, and e the identity element. The *order* of a is the smallest integer n such that:

$$\underbrace{a \circ a \circ \cdots \circ a}_{n} = e. \tag{10}$$

The set $\{a, a^2, \ldots, a^n\}$ (or $\{a, 2a, \ldots, na\}$) forms a *cyclic subgroup* of G of order n, with a as its generator. ◁

***Definition** 11.* A *field* is an algebraic structure composed of a set $\mathbb{F}$ together with the binary addition and multiplication operations, $+$ and $\times$, satisfying the following: $\mathbb{F}$ is an abelian group under addition $+$; $\mathbb{F}\backslash\{0\}$ is an abelian group under multiplication $\times$; $\times$ is distributive over addition: $a \times (b + c) = a \times b + a \times c$. A field is *finite* if $|\mathbb{F}| < \infty$. ◁

Examples of fields are the set of rational, real, complex numbers $\mathbb{Q}, \mathbb{R}, \mathbb{C}$. A fundamental field in cryptography, used in this paper, is the set of integers modulo l, $\mathbb{Z}/l\mathbb{Z}$, with l prime. We use the notation $\mathbb{F}_l = \mathbb{Z}/l\mathbb{Z}$ for compactness, not to be confused with the set of l-adic integers.

3.2 Elliptic curves

Given the preliminary definitions in Section 3.1, we can now introduce elliptic curves. an elliptic curve $E(\mathbb{F})$ is the set of points in a field $\mathbb{F}$, satisfying

$$y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5. \tag{11}$$

The Weierstrass normal form, defined as:

$$y^2 = x^3 + ax + b, \tag{12}$$

with $a, b \in \mathbb{F}$, is a special elliptic curve commonly used in cryptography. In Figure 2 we present an example of an elliptic curve defined for $\mathbb{F} = \mathbb{R}$, with $a = 1, b = -1$.

The curve $E(\mathbb{F})$ forms an abelian group together with an addition operator $+$, with the so called point at infinity O as its identity. Specifically, addition in $E(\mathbb{F})$ satisfies: $P + O = O + P = P$; for $P, Q \in E(\mathbb{F})$, $P \neq \pm Q$, the point $R = P + Q$ is the point satisfying $P + Q - R = O$; if $P = (x, y) \in E(\mathbb{F})$, then the negative of P, $-P = (x, -y)$, is such that $P + (-P) = O$. Coordinates $(x_R, y_R) = R = P + Q$ can be computed via:

$$\begin{cases} x_R = \lambda^2 - x_P - x_Q \\ y_R = \lambda(x_P - x_R) - y_P \end{cases}, \ \lambda = \frac{y_Q - y_P}{x_Q - x_P} \tag{13}$$

with $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$. Geometrically this corresponds to taking the line through P and Q and finding where it intersects the elliptic curve. Given operator $+$, we are also interested in the so called *doubling operator*, i.e., computing $2P = P + P$, which can be interpreted geometrically by taking the tangent to P, which crosses the elliptic curve in $-2P$. Then $2P$ is found by inverting the y coordinate. Then, coordinates (x_{2P}, y_{2P}) are defined:

$$\begin{cases} x_{2P} = \lambda^2 - 2x_P \\ y_{2P} = \lambda(x_P - x_{2P}) - y_P \end{cases}, \ \lambda = \frac{3x_P^2 + a}{2y_P}. \tag{14}$$

Geometric representations for addition and doubling are given in Figure 2. We can also define *scalar multiplication* over $E(\mathbb{F})$, by repeatedly adding a point $P \in E(\mathbb{F})$ to itself:

$$sP = \underbrace{P + P + \cdots + P}_{s}, \tag{15}$$

with $s \in \mathbb{Z}_+$. For elliptic curves over finite fields, used in Algorithm 1, this operation can be efficiently computed in $O(\log(s))$ using the *Double and Add* algorithm

(Wohlwend, 2016). Scalar multiplication can also be used to define a cyclic subgroup with generator P and order n.

***Definition** 12.* The order of a point $P \in E(\mathbb{F})$ is defined as the minimum $n_P \in \mathbb{Z}_+$ such that: $\underbrace{P + \cdots + P}_{n_P} = O$. ◁

***Definition** 13.* The cofactor of a point $P \in E(\mathbb{F}_p)$ of order n_P is defined as: $h_P \doteq |E(\mathbb{F})|/n_P$ ◁

Lagrange's theorem states that the order of a subgroup must be a divisor of the order of the group: thus, $h_P \in \mathbb{Z}_+$, for all $P \in E(\mathbb{F}_p)$ (Birkhoff and Mac Lane, 2017).

We have described elliptic curves on a generic field $\mathbb{F}$; however, in the following we are interested in elliptic curves defined on finite fields, and specifically $\mathbb{F}_p$, an example of which is found in Figure 3. This field is used also in ECC.

3.3 Elliptic curve cryptography

Similar to Diffie-Hellman-Merkle private key cryptography, ECC is based on the difficulty of solving the discrete logarithm problem on elliptic curves, i.e., even if P and $S = lP$ are known, there are no efficient ways to find l.

The Diffie-Hellman key exchange on an elliptic curve has the following structure: suppose Alice and Bob want to share a key, without any shared secrets. They agree on the parameters of a shared (public) elliptic curve $E(\mathbb{F}_p) = \{\mathbb{F}_p, P, n_P, h_P, a, b\}$, where $\mathbb{F}_p$ is the field on which the elliptic curve is defined, P is a generator point, n_P and h_P are respectively the order and cofactor of P, and a, b are the parameters of the elliptic curve (12). Alice defines a private key k_a, and computes $Q_a = k_a P$, her public key. Similarly, Bob defines private and public keys k_b and Q_b. Thus, by exchanging their public keys, Alice and Bob compute the same shared key, $Q_{ab} = k_a Q_b = k_b Q_b = (k_a k_b)P$. This key is created without sharing private information, and therefore any agent capable of eavesdropping the communication Alice and Bob, although knowing P, Q_a, and Q_b, cannot reconstruct k_a or k_b, as this would require solving a discrete logarithm problem on elliptic curves.

***Remark** 3.* The order and cofactor of a generator point P play a role in how difficult it is to compute the discrete logarithm problem. For cryptographic problems, P is often defined such that $h_P \leq 4$ (Wohlwend, 2016). ◁

4. SWITCHING FUNCTION DESIGN

Let us now present the algorithm that defines the switching functions σ. We drop subscripts q and w to improve readibility. We then highlight the information to be shared between $\mathcal{W}$ and $\mathcal{Q}$ for $\sigma_w = \sigma_q$.

4.1 An elliptic-curve-based switching function

Recall from (7) that $\sigma(\cdot)$ plays a role in changing the parameters of the watermarking pair, once the triggering function $\tau(y_p(\kappa)) = 1$, for $\kappa \in \mathcal{K}$. Here, we take $y_p(\kappa-1)$ as the input to the switching function $\sigma(\cdot)$. In the following, we suppose that $n_y = 1$, and therefore that $y_p \in \mathbb{R}$. The procedure can be easily extended for $n_y > 1$.

***Remark** 4.* It is necessary to take $y_p(\kappa - 1)$ rather than $y_p(\kappa)$ because $y_p(\kappa - 1) = y_q(\kappa - 1) \in \mathcal{I}_w \cap \mathcal{I}_q$, while $y_p(\kappa) \notin \mathcal{I}_q$, although $y_p(\kappa) \in \mathcal{I}_w$. Indeed, the watermark remover $\mathcal{Q}$ requires knowledge of the new parameter vector $\theta^+(\kappa)$ to recover the plant's measured output. ◁

***Remark** 5.* The switching function depends on y_p to introduce some randomness to the algorithm, thus making it more complex to identify for an attacker. ◁

The switching function $\sigma(\cdot)$ is the result of the following: the projection of $y_p(\kappa - 1)$ onto $P \in E(\mathbb{F}_s)$, the elliptic curve defined on the field of integers modulo s, with s prime; the computation of $S = lP$, $S \in E(\mathbb{F}_s)$, for some $l \in \mathbb{Z}_+$; the mapping of S onto Θ, the constrained parameter set guaranteeing that the resulting watermarking pair $\{\mathcal{W}, \mathcal{Q}\}$ satisfies conditions in Definition 1.

In the first step the measurement output is "projected" to a point on the elliptic curve $E(\mathbb{F}_s)$, then used as the generator of a cyclic subgroup. We define function $\alpha(\cdot) : \mathbb{R}^{n_y} \to E(\mathbb{F}_s)$ as this function. In turn $\alpha(\cdot)$ can be seen as $\alpha(\cdot) = \alpha_2(\alpha_1(\cdot))$, where $\alpha_1(\cdot) : \mathbb{R}^{n_y} \to \mathbb{R}_s \times \mathbb{R}_s$ scales the output, with $\mathbb{R}_s$ the reals modulo s, defining the coordinates of $\tilde{P} = (x_{\tilde{P}}, y_{\tilde{P}}) \in \mathbb{R}_s \times \mathbb{R}_s$, whilst $\alpha_2 : \mathbb{R}_s \times \mathbb{R}_s \to E(\mathbf{F}_s)$ maps the point $\tilde{P}$ to $P \in E(\mathbb{F}_s)$. Specifically $\alpha_1 = (\alpha_{1,x}, \alpha_{1,y})$,where $\alpha_{1,x}$ and $\alpha_{1,y}$ may have similar structure, but should have different parameters, such that $x_{\tilde{P}} \neq y_{\tilde{P}}$, in general. These functions may be defined in several ways, e.g., that given in Section 5. Once $\tilde{P}$ is computed, $\alpha_2(\cdot) : \mathbb{F}_s \times \mathbb{F}_s \to E(\mathbb{F}_s)$ can be defined as:

$$P = \alpha_2(\tilde{P}) = \underset{\Lambda \in E(\mathbb{F}_s)}{\arg\min} \|\Lambda - \tilde{P}\|_2^2, \tag{16}$$

thus defining a generator point $P \in E(\mathbb{F}_s)$.

***Remark** 6.* Scaling the measurement output y_p is a critical step of the switching function. Whilst in steady state, the output of the plant varies in a (likely small) neighborhood of some nominal output, r, which may be given by a reference to the controller. Thus, without scaling by $\alpha_1(\cdot)$, the result of the projection of y_p would always be onto a subset $\mathcal{E} \subset E(\mathbb{F}_s)$, with $|\mathcal{E}| << |E(\mathbb{F}_s)|$, and possibly $|\mathcal{E}| = 1$. This is undesirable, as it implies that $\theta^+ = \theta$. ◁

Once a generator point $P \in E(\mathbb{F}_s)$ is computed, a scalar multiplication is performed to define $S = lP$. Given the definition of time-varying generator P, it may be that for some values of P, l is the order of P, and thus that $S = O$. Thus, if $S = O$, some heuristic can be selected, such as $S = P$ or $S = -P = (l-1)P$, to avoid the result $S = O$.

***Remark** 7.* Note that here we assume l to be defined a priori and to be time invariant. This is done because, if l were time-varying, it would either have to be a function of $y_p(\kappa - 1)$, or to satisfy another switching function $l^+ = \pi(l)$. While the former is undesirable because it may expose the switching function to attacks capable of reconstructing y_p, the latter would require the definition of another switching function. ◁

Finally, $\eta(\cdot) : E(\mathbb{F}_s) \to \Theta$ is introduced to map the value of S to an appropriate parameter vector, ensuring that the resulting watermarking pair satisfies Definition 1. Similarly to $\alpha(\cdot)$, $\eta(\cdot)$ may also be seen as the composition of two functions, $\eta(\cdot) = \eta_2(\eta_1(\cdot))$, where η_1 maps the points of the elliptic curve into $\mathbb{R}^{n_\theta}$, and η_2 restricts the novel parameter vector to satisfy $\theta^+ \in \Theta$. Once again, in order to ensure that changes in S lead to large differences in θ^+,

Algorithm 1 Switching function $\sigma_w(y_p(\kappa_w - 1))$

1: **Input:** $y_p(\kappa_w - 1)$, $E(\mathbb{F}_q)$, l, $\alpha(\cdot)$, $\eta(\cdot)$;
2: **Output:** θ_w^+
3: Compute the generator of the elliptic curve by computing $P = \alpha(y_p(\kappa_w - 1))$;
4: Given P, compute $S = lP$;
5: Define $\theta_w^+(\kappa_w) = \eta(S)$
6: **return:** $\theta_w^+(\kappa_w)$

$\eta_1(\cdot) : E(\mathbb{F}_s) \to \mathbb{R}^{n_\theta}$ should be chosen nonlinear, while the definition of $\eta_2(\cdot) : \mathbb{R}^{n_\theta} \to \Theta$ depends on the class of systems that are used to define $\mathcal{W}$ and $\mathcal{Q}$. In Section 4.3 we give an example of how to construct $\eta_2(\cdot)$ for the class of finite impulse response (FIR) filters.

Finally, let us comment on what information must be included in $\mathcal{I}_w$ and $\mathcal{I}_q$ to ensure that $\sigma_w(\cdot) = \sigma_q(\cdot) = \sigma(\cdot)$, and thus that $\theta_w^+ = \theta_q^+$. Recall from Remark 4 that $y_p(\kappa - 1) \in \mathcal{I}_w \cap \mathcal{I}_q$, for all $\kappa \in \mathcal{K}$, provided $\mathcal{K}_w = \mathcal{K}_q$, and $\{\mathcal{W}, \mathcal{Q}\}$ satisfy Definition 1. Thus, it is necessary that

$$\{\alpha(\cdot), l, E(\mathbb{F}_s), \eta(\cdot)\} \subset \mathcal{I}_w \cap \mathcal{I}_q. \tag{17}$$

This information is time invariant, to be shared securely at the initial design time of the watermarking system.

***Remark** 8.* In practical applications, attention must be given to the definition of $\alpha(\cdot)$ and $\eta(\cdot)$, to guarantee that numerical issues do not cause errors in the computation of the watermarking parameters, and therefore that $\theta_w^+ \neq \theta_q^+$. In-depth analysis of this phenomenon, and methods to solve it are, however, outside of the scope of this paper and will be the subject of future research. ◁

4.2 Security analysis

Let us now evaluate the security of the switching function $\sigma(\cdot)$, as defined in Section 4.1. Here, we consider "security" of $\sigma(\cdot)$ to mean that it is not possible for an attacker to predict, at time $\kappa \in \mathcal{K}$, what the value of θ^+ is. Indeed, even if an attacker had knowledge of θ, and were therefore capable of constructing an undetectable attack, without knowledge of θ^+ its undetectability would be threatened.

In order to formally examine this scenario, let us introduce the set of information known by the attacker, $\mathcal{I}_a(k), k \geq K_a$. This, similarly to the definitions of $\mathcal{I}_w(k)$ and $\mathcal{I}_q(k)$, is the set of all signals and parameters that are known to the malicious agent at time k.

To be able to compute θ^+, it is necessary to know $\sigma(\cdot)$, and therefore necessary that:

$$\{\alpha(\cdot), \eta(\cdot), l, E(\mathbb{F}_s)\} \subseteq \mathcal{I}_a. \tag{18}$$

What is particularly interesting, and a direct consequence of using elliptic curves, is that even if all functions were known to the attacker, so long as $l \notin \mathcal{I}_a$, the attacker would not be capable of reconstructing θ^+. Indeed, even if an eavesdropping attacker were to be able to estimate θ and θ^+, after some delay following the switch, and through knowledge of $\alpha(\cdot)$ and $\eta(\cdot)$ were to reconstruct P and S, finding l solving $S = lP$ is the solution to the discrete logarithm over elliptic curves.

4.3 Watermark pair stability: FIR filters

Let us now focus on the definition of the parameter set Θ such that if $\theta_w^+ \in \Theta$, the resulting watermarking pair is guaranteed to satisfy the conditions in Definition 1. We restrict the set of parameters by giving the watermarking generator and remover some *structure*, namely, for the purpose of this paper, we suppose that $\mathcal{W}$ is composed of n_y *parallel* FIR filters of order n_h; thus, $n_\theta = n_y \cdot n_h$. For the sake of maintaining notation streamlined, and without loss of generality, for the remainder of this subsection we suppose that $n_y = 1$. The output of $\mathcal{W}$ in (3a) is thus:

$$y_w(k) = \sum_{h=0}^{n_h} b_h y_{p,i}(k - i), \tag{19}$$

with $b_h \in \mathbb{R}, \forall h \in \{0, 1, \ldots, n_h\}$. This formulation guarantees that $\mathcal{W}$ is stable, with n_h poles at the origin. Finding $\Theta \subseteq \mathbb{R}^{n_h}$ is equivalent to finding the set of parameters b_h for which $\mathcal{W}$ is invertible, with stable inverse.

***Theorem** 1.* Suppose that $\mathcal{W}$ is an LTI system with dynamics defined by the FIR filter in (19). Thus, if

$$b_0 \neq 0\,, \quad |b_1| < 1\,, \quad \sum_{i=2}^{n_h} \left|\frac{b_i}{b_0}\right| < 1 - |b_1| \tag{20}$$

hold, the resulting watermarking pair $\{\mathcal{W}, \mathcal{Q}\}$ is guaranteed to satisfy Definition 1, with $\theta_w = \theta_q = \mathrm{col}_i[b_i]$. □

5. NUMERICAL RESULTS

The elliptic curve we choose for this example is that defined on $\mathbb{F}_{17}$, shown in Figure 3. This is an elliptic curve, of order $|E(\mathbb{F}_{17})| = 19$, on which the solution to the discrete logarithm problem does not require a lot of computation; indeed, in cryptographic settings, FIPS 186-4 recommends using elliptic curves defined on fields where the prime integer used for the modulo operation is at least 192 bits long. However, its structure makes it suitable for illustrating some fundamental characteristics of $\sigma(\cdot)$. Because its order is a prime number, the order of each of its points $P \in E(\mathbb{F}_{17})$ is also 19, as the coefficient h of P, defined in Definition 13, is always an integer (Birkhoff and Mac Lane, 2017). Furthermore, by selecting l such that $l \bmod 19 \neq 0$, it is possible to guarantee that $S = lP \neq O$, the point at infinity, for all $P \in E(\mathbb{F}_{17})$. Moreover, the order of $E(\mathbb{F}_{17})$ also determines the number of parameters $\theta_w = \theta_q \in \Theta$ that define $\{\mathcal{W}, \mathcal{Q}\}$. It is worth noting that the points $P \in E(\mathbb{F}_{17})$ do not partition the space uniformly: to show this, in Figure 3 we include the Voronoi diagram generated by taking the points in $E(\mathbb{F}_{17})$ as seeds.

We can now focus our attention on the definition of $\alpha(\cdot)$ and $\eta(\cdot)$, supposing $n_y = 1$. As discussed previously, both of these can be seen as the combination of a scaling and a projection function. While the latter have been discussed in Section 4.1, a possible definition of the former is:

$$\alpha_{1,i}(\gamma) = a_{i,0}\mathrm{atan}(a_{i,1}\gamma) + \sum_{j=2}^{n_\alpha} a_{i,j}|\gamma|^j \tag{21}$$

$$\eta_1(H) = \sum_{j=0}^{n_\eta} b_j \|H\|_2^j \tag{22}$$

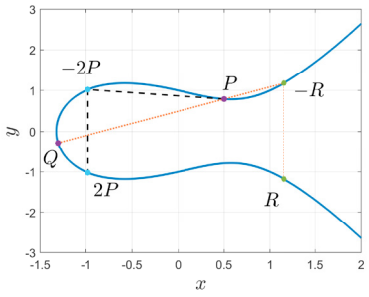

Fig. 2. The elliptic curve defined in $\mathbb{R}$, with parameters $a = -1$ and $b = 1$, with geometric representation of the operations $R = P + Q$ and $2P$.

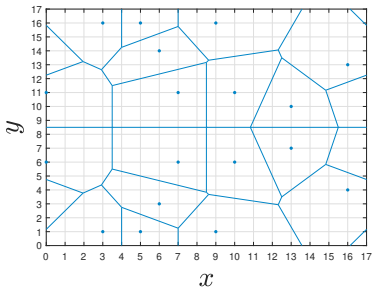

Fig. 3. The points of the elliptic curve in $\mathbb{F}_{17}$, with $a = 2$ and $b = 2$, together with the Voronoi partition taking these points as seeds.

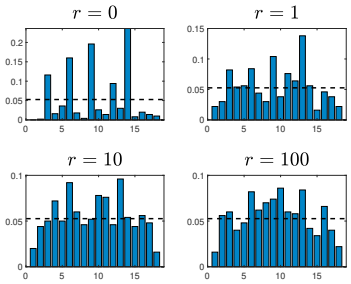

Fig. 4. Relative frequency of a point being reached from $\alpha_1(y)$, for each point in the elliptic curve $E(\mathbb{F}_{17})$, indexed over $\{1, \ldots, |E(\mathbb{F}_{17})|\}$.

with $H \in E(\mathbb{F}_{17})$, and where $a_{i,j}, i \in \{x, y\}, j \in \{0, 1, \ldots, n_\alpha\}$ and $b_j, j \in \{0, 1, \ldots, n_\eta\}$ are time-invariant parameters, shared between $\mathcal{W}$ and $\mathcal{Q}$.

Finally, we are interested in examining the sensitivity of the output of $\sigma(\cdot)$ with respect to (small) variations of the plant measurement outputs y_p. Results are presented in Figure 4. We consider the measurement of the plant $y_p = r + \epsilon$, where $r \in \mathbb{R}^{n_y}$ is an output reference for $\mathcal{P}$, and ϵ is the tracking error. Specifically, we suppose $r \in \{0, 1, 10, 100\}$, and for each we consider 500 realizations of ϵ, each taken from a uniform distribution with limits $[-0.05, 0.05]$. In Figure 4 we show how frequently different points $P \in E(\mathbb{F}_{17})$ are reached for changes in y_p, relative to the total number of values of ϵ taken. We see that, irrespective of the operating point, it is possible to reach all cells, which implies that for all operating conditions all 19 values that θ^+ can take are reachable. However, there are large differences between the behaviors relative to the reference points themselves. Indeed, for $r = 0$, it is significantly more likely that a subset $\mathcal{E}$ of all points in the elliptic curve is reached, with $|\mathcal{E}| = 5$. On the other hand, for $r = 10$ the probability of reaching a given point in the elliptic curve is closer to being uniform across all points. In this case, $\sigma(\cdot)$ is more sensitive to small changes in the input. This gives us important insight for the practical implementation of σ: indeed, $\alpha_1(\cdot)$, which is the main "driver" of this sensitivity, is to be appropriately tuned for those points that are to be tracked by the plant, to ensure there is sufficient sensitivity of the switching function $\sigma(\cdot)$.

6. CONCLUSION

In this work, we have presented a method to define the switching function for switching multiplicative watermarking based on elliptic curves. We show how, even for attackers with large amounts of information, the switching signal may remain secure. In future work, we are interested in testing the real-world applicability of this scheme, testing it on real world hardware.

REFERENCES

Birkhoff, G. and Mac Lane, S. (2017). *A survey of modern algebra*. CRC Press.

Chen, Z., Pasqualetti, F., He, J., Cheng, P., Trentelman, H.L., and Bullo, F. (2020). Guest editorial: Special issue on security and privacy of distributed algorithms and network systems. *IEEE Trans. on Autom. Control*, 65(9), 3725–3727.

Falliere, N., Murchu, L.O., and Chien, E. (2011). W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5(6), 29.

Ferrari, R.M. and Teixeira, A.M. (2020). A switching multiplicative watermarking scheme for detection of stealthy cyber-attacks. *IEEE Trans. on Autom. Control*.

FIPS 186-4 (2013). Digital Signature Standard (DSS). Standard, National Institute of Standards and Technology.

Gallo, A.J. and Ferrari, R.M.G. (2022). Cryptographic switching functions for multiplicative watermarking in cyber-physical systems. URL `https://arxiv.org/abs/2203.11851`.

Griffioen, P., Weerakkody, S., and Sinopoli, B. (2020). A moving target defense for securing cyber-physical systems. *IEEE Trans. on Autom. Control*, 66(5), 2016–2031.

Lee, E.A. (2008). Cyber physical systems: Design challenges. In *ISORC 2008*, 363–369.

López, J. and Dahab, R. (2000). An overview of elliptic curve cryptography. Technical report.

Sandberg, H., Amin, S., and Johansson, K.H. (2015). Cyberphysical security in networked control systems: An introduction to the issue. *IEEE Control Systems*, 35(1), 20–23.

Weerakkody, S., Ozel, O., Mo, Y., Sinopoli, B., et al. (2019). Resilient control in cyber-physical systems: Countering uncertainty, constraints, and adversarial behavior. *Foundations and Trends® in Systems and Control*, 7(1-2), 1–252.

Weerakkody, S. and Sinopoli, B. (2015). Detecting integrity attacks on control systems using a moving target approach. In *2015 54th IEEE Conf. on Decision and Contr. (CDC)*, 5820–5826. IEEE.

Wohlwend, J. (2016). Elliptic curve cryptography: Pre and post quantum. Technical report, Technical report.

Zhang, X., Polycarpou, M.M., and Parisini, T. (2002). A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Trans. on Autom. Control*, 47(4), 576–593.