An Impact-Aware Defense against Stuxnet

Andrew Clark, Quanyan Zhu, Radha Poovendran and Tamer Başar

***Abstract*—The Stuxnet worm is a sophisticated malware designed to sabotage industrial control systems (ICSs). It exploits vulnerabilities in removable drives, local area communication networks, and programmable logic controllers (PLCs) to penetrate the process control network (PCN) and the control system network (CSN). Stuxnet was successful in penetrating the control system network and sabotaging industrial control processes since the targeted control systems lacked security mechanisms for verifying message integrity and source authentication. In this work, we propose a novel proactive defense system framework, in which commands from the system operator to the PLC are authenticated using a randomized set of cryptographic keys. The framework leverages cryptographic analysis and control- and game-theoretic methods to quantify the impact of malicious commands on the performance of the physical plant. We derive the worst-case optimal randomization strategy as a saddle-point equilibrium of a game between an adversary attempting to insert commands and the system operator, and show that the proposed scheme can achieve arbitrarily low adversary success probability for a sufficiently large number of keys. We evaluate our proposed scheme, using a linear-quadratic regulator (LQR) as a case study, through theoretical and numerical analysis.**

I. Introduction

Industrial control systems (ICSs) are ubiquitous in applications ranging from material processing to power generation and transmission. Such systems increasingly rely on remote operations via local area networks or the Internet, which are enabled by software with limited security protections. As a result, ICSs are inviting targets for adversaries who attempt to disable critical infrastructure through cyber attacks.

The threat of cyber attacks on ICS was demonstrated by the Stuxnet worm, which exploited several previously unknown vulnerabilities in the Windows operating system and Siemens STEP 7 software to target specific control systems appearing in uranium enrichment facilities [1], [2]. Stuxnet-type malware targets the control system by compromising workstations used to reconfigure the Programmable Logic Controllers (PLCs) for facility operations and tampering with messages sent from the system operator to the PLC. For example, Stuxnet modifies control messages in order to increase the frequency of nuclear centrifuges to unsafe levels, leading to equipment failure [3]. The appearance of Stuxnet has led to research into vulnerabilities of ICS software [4], as well as introducing new security checks into control software and hardware [5], [6]. In particular, control messages must be secured in order to ensure the authenticity of the source, detect modification of received messages, and prevent replay attacks.

In the security literature [7], [8], these properties are provided efficiently using cryptographic mechanisms, such as message authentication codes. Securing communication between the operator and PLC can be achieved by either using the same cryptographic key for all messages, or using a different key for each message. Under both approaches, however, tampering with any message requires constant workload for the adversary, even though the impact of a tampered message varies based on its effect on the physical plant.

In this paper, we introduce a framework for securing industrial control systems against Stuxnet-type malware by incorporating the impact of tampered message on system performance into the defense strategy. We observe that there are two possible diversity-based proactive cryptographic defense schemes within this solution framework: (a) increasing the cryptographic key length so that high-impact messages are more computationally difficult to tamper, and (b) given a set of keys, choosing the subset of messages authenticated by each key so that the expected impact on the physical system of compromising any one key is minimized.

Since the key lengths are typically fixed by common security standards [9] and cryptographic algorithms are often optimized for a fixed set of key lengths, we focus on the latter scheme in this work. We make the following specific contributions:

- We propose a system architecture for securing the communications between a system operator and PLCs against Stuxnet-type malware. We take into account the cyber aspect, including authentication of control messages to the PLCs, as well as the physical aspects, including the damage caused by unauthorized messages on ICSs.
- We formulate the key management problem within a convex optimization framework for minimizing the expected damage on ICSs and provide efficient solution algorithms. We show that the defense gain, defined as the ratio between the impact of the attack with non-proactive and proactive defenses, increases exponentially as a function of the key length and polynomially in the number of message classes.
- As a case study, we evaluate the proposed mechanism for protecting an industrial plant modeled by a linear quadratic regulator, in which an adversary aims to modify the set point values for the regulator, and derive the weights representing the impact of the attack from compromising each message class.
- Our results are corroborated through a simulation study, in which we show that a significant decrease in the impact of the attack can be provided by randomizing between

The research was partially supported by the AFOSR MURI Grant FA9550-10-1-0573, ARO Grant W991NF-12-1-0448, and also by an NSA Grant through the Information Trust Institute at the University of Illinois.

Andrew Clark and Radha Poovendran are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA. Email: {awclark, rp3}@u.washington.edu

Quanyan Zhu and Tamer Başar are with the Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. Email: {zhu31, basar1}@illinois.edu

a small set of keys. We illustrate how the effect of the attack changes depending on which component of the set point is modified, and discuss how to provide additional protection for more sensitive messages within our framework.

The paper is organized as follows. Section II reviews the related work. Section III provides background on ICSs, PLCs and the Stuxnet worm. Section IV defines the system and adversary model and introduces security metrics. Section V presents problem formulation for selecting the set of messages authenticated using each key. Section VI describes a case study based on LQR control. Section VII presents simulation results. Section VIII concludes the paper.

II. Related Work

Since the advent of the Stuxnet virus, there has been significant research into identifying vulnerabilities in the software used to program PLCs. Discussions of the Stuxnet worm can be found in [3], while a broader review of topics in PLC security is given in [4]. Cyber security requirements for industrial control systems are given in [9]. These existing models focus on identifying and removing software vulnerabilities, however, rather than establishing a broader analytical framework for cyber security of ICS.

In the control-theoretic community, attacks in which an adversary compromises one or more sensors in order to inject false data have been considered [10]. This attack model differs from our case, in which an adversary attempts to compromise the device used to program the controller, and hence is complementary to our line of research.

Vulnerabilities of the operating systems used by embedded control systems have been identified as a potential security threat [11]. While efforts at designing secure operating systems for these embedded systems are underway [12], to the best of our knowledge no such operating system specially designed for secure control has been released.

Proactive, diversity-based defense mechanisms, in which the system randomizes its internal state in order to reduce the impact of attacks, are an emerging area in the cyber security community. For example, address space randomization, in which the address of a device is randomized in order to thwart code injection attacks, has been deployed in modern operating systems including Windows, Linux, and iOS [13]. Such defense techniques have not yet been applied to the domain of cyber-physical systems for designing security mechanisms to protect control systems in modern critical infrastructures.

The impact of cyber attacks on physical systems is a major concern in control systems. The design of cyber defense needs to take into account the effect of communication delay and packet loss [15], the design specifications on the robustness and resilience of the system [16], and its information and system architecture [17]. Following these principles, the goal of this paper is to design a proactive defense mechanism using cryptographic solutions at the interface between the cyber and physical components of industrial control systems.

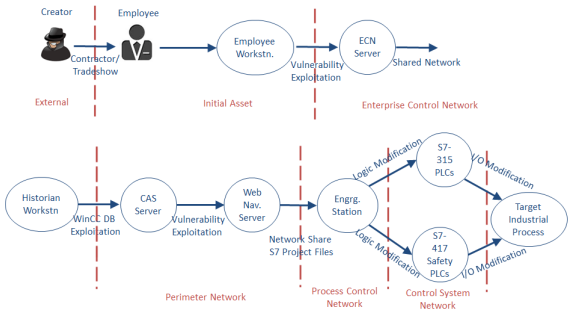


Fig. 1. An example sequence of attacks in Stuxnet adapted from [2]

III. Background on Stuxnet Malware

The Stuxnet worm leverages known and previously unknown vulnerabilities to install, infect and propagate, aiming to sabotage industrial processes operated by Siemens SIMATIC WinCC and PCS 7 control systems [2], [3]. Figure 1 uses an example sequence of attacks to illustrate how the worm propagates through the enterprise control network (ECN) to the control system network (CSN). The worm first propagates via infected removable drives (such as flash drives and external portable hard disks), and then local area network communications (such as shared network drives and print spooler services), and finally infects Siemens project files, including both WinCC and STEP 7 files, which are used to program the PLC.

In conventional ICS network architectures, the process control network (PCN) and control system network (CSN) are hosted in the same security zone [2], [18]. The PCN hosts plant operators on their human machine interface (HMI) workstations. The CSN is dedicated to traffic specifically related to automation and control such as traffic to and from PLCs. This has created potential security hazards for CSN once the worm penetrates the perimeter network and PCN since no firewalls are used to separate the two networks.

Figure 2 illustrates the interactions between PCN and CSN. The connection between the PCN and CSN is managed by a library file, which calls different routines to read and write to memory on the PLC. By replacing the library files, Stuxnet can tamper with commands from the PCN without being detected by the PLC or system operator, since there are no integrity checks used to verify the source of a message [3]. In the 2010 Iranian Natanz nuclear facility incident [1], a function block `DP_RECV` for receiving network frames on the Profibus, a standard industrial network bus used for distributed I/O, is replaced by a malicious block. Each time the function is used to receive a packet, the malicious Stuxnet block takes control and does post-processing on legitimate packet data, and hence affect the PLC and the control system.

IV. System and Adversary Model

In this section, we describe the system and adversary models, including the capabilities of the adversary and the defense mechanisms employed by the system. We then define the security metrics considered in this work.

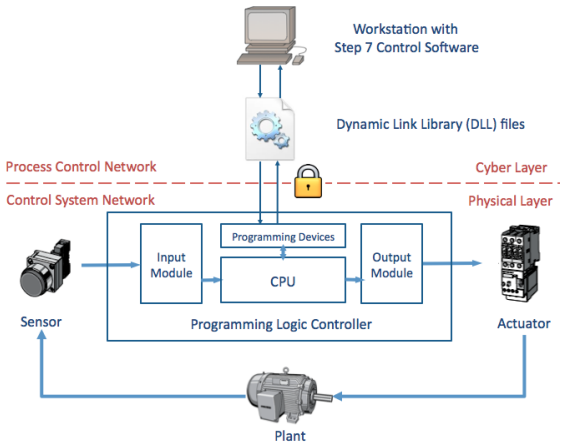


Fig. 2. Interactions between process control network (PCN) and control system network (CSN) in industrial control systems (ICSs): A proactive cryptographic solution is proposed at the information exchange interface between PCN and CSN.

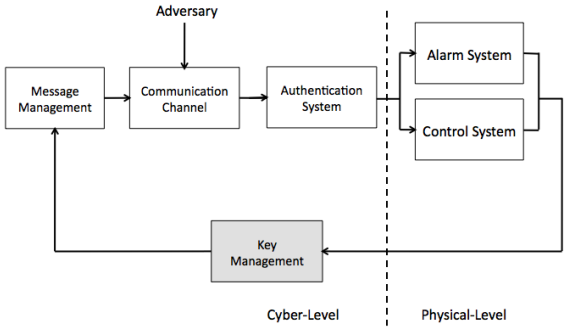


Fig. 3. A system model of the cryptographic solution for protecting ICSs from cyber attacks.

A. System Model

The cryptographic solutions to ICSs against Stuxnet-type worms can be described by the system model described in Figure 3. The model constitutes 6 building blocks. A workstation operator sends messages to PLCs through a communication channel that can be subject to attacks. The message management categorizes messages in different classes including dummy messages that are used for deceiving the adversary. The receiver end executes the message if it is legitimately authenticated. The operator sends commands to the PLC from a message space $\mathscr{M}$. The set $\mathscr{M}$ is divided into n classes, denoted $\mathscr{M}_1,\ldots,\mathscr{M}_n$, with $\mathscr{M}=\cup_{i=1}^{n}\mathscr{M}_i$ and $\mathscr{M}_i\cap\mathscr{M}_j=\emptyset$ for $i\neq j$. We group the message classes into two different types. One type is used for standard command and control. The second type of message classes contain dummy messages that are used for deception, which results in no PLC responses. The set of indices $\mathscr{R}\subseteq\{1,\ldots,n\}$ correspond to classes of messages that overwrite the internal logic of the PLC. Messages in the set of indices $\mathscr{F}=\{1,\ldots,n\}\setminus\mathscr{R}$ are ignored by the PLC, and are used only to deceive the adversary. We assume that the fraction of messages in each class can be varied. Varying the number of messages per class will affect the cryptographic computations performed by the operator and PLC, but not the control messages themselves.

To prevent an adversary from altering or injecting messages, we propose that a message authentication code (MAC) be appended to each message sent from the operator. The MAC is described by the hash function $h:\mathscr{M}\times\mathscr{K}\rightarrow\mathscr{Y}$, where $\mathscr{K}$ is the set of keys and $\mathscr{Y}$ is the set of hash outputs. All message classes use the same MAC function h. Let $p=|\mathscr{K}|$ represent the number of possible keys. Messages from set $\mathscr{M}_i$ are authenticated using key $K_i\in\mathscr{K}$, with $K_i\neq K_j$ for $i\neq j$.

The PLC authenticates the (message, MAC) pair (x,y) by consulting a predefined look-up table known to both the operator and the PLC, which identifies the set $\mathscr{M}_i$ with $x\in\mathscr{M}_i$. We assume that generating such a look-up table is feasible if the number of sensitive messages that must be authenticated is small. Otherwise, the operator and PLC can generate a table via a keyed hash function $h':\mathscr{M}\times\mathscr{K}\rightarrow\{1,\ldots,n\}$.

The PLC then checks if $h(x,K_i)=y$. Messages that fail this authentication check are discarded and ignored. Furthermore, if $x\in\mathscr{M}_i$ and $h(x,K_j)=y$ with $K_j\neq K_i$, then the message is identified as a possible forgery attempt and a warning is triggered by an alarm system depicted in Fig. 3.

It is assumed that the contents of the messages are not encrypted, as the operator gives higher priority to ensuring that no forged messages are inserted (i.e., guaranteeing message integrity) than to protecting against passive eavesdropping.

B. Adversary Model

We consider an active adversary, who is capable of eavesdropping on messages exchanged between the operator and PLC, sending (message, MAC) pairs (x,y) to the PLC, and performing computations using probabilistic polynomial-time algorithms. Let $\mathbf{Alg}_0$ denote the set of feasible algorithms for determining the keys $K_1,\ldots,K_n$. The goal of the adversary is to insert a (message, MAC) pair (x,y) such that $h(x,K_i)=y$ and $x\in\mathscr{M}_i$.

Assumption 1: The adversary knows the number of message classes, n. However, given a valid (message, MAC) pair (x,y), an adversary who does not know any of the keys $K_1,\ldots K_n$ cannot determine whether $x\in\mathscr{M}_i$ for any $i=1,\ldots,n$.

The assumption that the adversary does not know the mapping between messages and keys is justified by the fact that interactions between the operator and PLC take place intermittently in a closed environment, giving the adversary few opportunities to eavesdrop on messages. This assumption holds for the Stuxnet malware, although secure storage techniques should be used to hide the message/key mapping from more sophisticated adversaries.

Assumption 2: The adversary has a probabilistic polynomial-time algorithm $\mathbf{Adv}_0\in\mathbf{Alg}_0$ that takes as input a set of q (message, MAC) pairs $(x_1,y_1),\ldots,(x_q,y_q)$ signed by a key $K\in\mathscr{K}$. The algorithm outputs a key $\tilde{K}$; we define $f(p,q):=\mathbf{Pr}(\tilde{K}=K)$. Furthermore, if there exist i, j, r, and s with $i\neq j$ and $r\neq s$, such that $y_i=h(x_i,K_r)$ and $y_j=h(x_j,K_s)$, $x_i\in\mathscr{M}_r$, and $x_j\in\mathscr{M}_s$, then $\mathbf{Pr}(\tilde{K}\in\{K_1,\ldots K_n\})=0$.

Assumption 2 implies that, if the adversary inputs two (message, MAC) pairs with distinct keys into a cryptanalytic algorithm, then the algorithm will fail to return a correct key. An example of $\mathbf{Adv}_0$ for a particular class of hash functions is given in Section IV-D.

C. Security Metric Definitions

Security metrics quantify the probability that the adversary will succeed in compromising one or more keys and injecting false messages. We first define the success probability, denoted P_s, which is equal to the probability that the adversary compromises at least one key.

Definition 1 (Success probability): Let $\mathscr{A}$ denote the set of probabilistic polynomial-time algorithms calling $\mathbf{Adv}_0$ as a subroutine, and let $\mathbf{Adv} \in \mathscr{A}$. Let $P_s(p,q,M;\mathbf{Adv})$ denote the probability that $\mathbf{Adv}$ correctly computes at least one K_i after observing M messages. We define the adversary's success probability to be

$$P_s^*(p,q,M) = \max_{\mathbf{Adv}\in\mathscr{A}} P_s(p,q,M;\mathbf{Adv}). \tag{1}$$

D. Security Metric Analysis for Universal Hash Function

The following MAC construction, first appearing in [8], this is used in the subsequent analysis.

Definition 2: The strongly-universal hash function MAC takes as input messages in $\mathbb{Z}_p$, the integers modulo p where p is a prime. The key is a sequence of coefficients $b_0,\ldots,b_{q_0-1} \in \mathbb{Z}_p$. The MAC $\tilde{h}:\mathscr{M}\times\mathbb{Z}_p^{q_0}\to\mathbb{Z}_p$, which is a special form of h from Section IV-A is defined by

$$\tilde{h}(x,b_0,\ldots,b_{q_0-1}) = \sum_{i=0}^{q_0-1} b_i x^i \bmod p, \tag{2}$$

where x^i denotes x raised to the i^{th} power.

The following lemma gives the security analysis of this MAC when only one key is used.

Lemma 1: Under Assumptions 1 and 2, the best-possible probability of recovering the correct key for the MAC in Definition 2 is $f(p,q) = \left(\frac{1}{p}\right)^{(q_0-q)_+}$.

Proof: A proof can be found in [7, Ch 4]. ∎

We now analyze the security when multiple keys are used, as described in Section IV-A.

Lemma 2: Suppose that the adversary has access to M_i distinct (message, MAC) pairs from set $\mathscr{M}_i$ for $i = 1,\ldots,n$. Then

$$P_s^*(p,q,M) = \max_{q\in\mathbb{Z}_{q_0}} \left\{ \frac{1}{\binom{M}{q}} \sum_{i=1}^{n} \binom{M_i}{q} f(p,q) \right\}. \tag{3}$$

A proof is given in the appendix. Note that, since $f(p,q)=1$ for $q \ge q_0$, $P_s^*(p,q,M)$ is strictly decreasing as a function of q for $q > q_0$.

V. Problem Formulation

In this section, we introduce a model for the interaction between the cyber and physical components of the system and present our optimization approach for designing a proactive randomization defense. The goal of the defender is to minimize the damage to the ICS caused by the injection of false messages. This can be accomplished by optimizing the number of messages of each class, $M_1,\ldots,M_n$.

Let ω_i denote the fraction of messages from $\mathscr{M}$ contained in $\mathscr{M}_i$, so that $\omega_i = \frac{M_i}{M}$. The impact of the adversary's attack is defined to be the expected damage to the system from a successful forged message, under the assumption that the adversary attempts to forge each message with equal probability.

Let $a_{i,m}$ denote the damage to the plant from an adversary injecting message $m \in \mathscr{M}_i$. The value of $a_{i,m}$ depends on the physical plant model; in some cases, $a_{i,m} = 0$ if the message is not harmful. We assume that $a_{i,m}$ is constant in time. An example of the derivation of $a_{i,m}$ is given in Section VI. Since the forgery is successful if the adversary determines the key K_i with $m \in \mathscr{M}_i$, the probability of success for a given choice of q, denoted $s_m(q)$, is

$$s_m(q) := \frac{\binom{M_i}{q}}{\binom{M}{q}} \frac{M_i}{M} = \frac{\binom{\omega_i M}{q}}{\binom{M}{q}} \omega_i.$$

Define the attack impact function $g:\Delta^n\times\mathbb{Z}_M\to\mathbb{R}$

$$g(\omega_1,\ldots,\omega_n;q) = \sum_{i=1}^{n} a_i^* \frac{\binom{\omega_i M}{q}}{\binom{M}{q}} f(p,q)\omega_i,$$

where Δ^n is the n-dimensional simplex; $a_i^* = \frac{1}{|\mathscr{M}_i|}\sum_{m\in\mathscr{M}_i} a_{i,m}$, equal to the expected impact on the system from compromising a randomly chosen message in $\mathscr{M}_i$. We denote the worst-case damage to the system by $g^*(\omega_1,\ldots,\omega_n)$, given as

$$g^*(\omega_1,\ldots,\omega_n) = \max_{q\in\mathbb{Z}_{q^0}} \left\{ \sum_{i=1}^{n} a_i^* \frac{\binom{\omega_i M}{q}}{\binom{M}{q}} f(p,q)\omega_i \right\}.$$

The problem of minimizing the impact of the attack on the system performance is equivalent to selecting the probability distribution $(\omega_1,\ldots,\omega_n)$ that minimizes the expected cost $g(\omega_1,\ldots,\omega_n)$. The problem is formulated as

$$\begin{aligned} \underset{\omega_1,\ldots,\omega_n}{\text{minimize}} \quad & \max\left\{ \textstyle\sum_{i=1}^{n} a_i^* \frac{\binom{\omega_i M}{q}}{\binom{M}{q}} f(p,q)\omega_i : q \in \{1,\ldots,M\} \right\} \\ \text{s.t.} \quad & \omega_1 + \cdots + \omega_n = 1 \end{aligned} \tag{4}$$

Remark 1: Problem (4) can be interpreted as a zero-sum game between the system and adversary. The system selects a probability distribution $(\omega_1,\ldots,\omega_n)$ in order to reduce the impact of the attack, while the adversary selects q in order to maximize the impact of injecting a message and disrupting the system. □

The following proposition leads to efficient algorithms for computing $g^*(\omega_1,\ldots,\omega_n)$.

Proposition 1: For fixed $(\omega_1,\ldots,\omega_n)$, there exists a unique point $q^* \in \{1,\ldots,q_0\}$ such that $g(\omega_1,\ldots,\omega_n;q)$ is nondecreasing for $1 \le q \le q^*$ and nonincreasing for $q \ge q^*$. Furthermore, $q^* \in \arg\max\{g(\omega_1,\ldots,\omega_n;q) : q \in \{1,\ldots,M\}\}$.

A proof is given in the appendix. In order to solve (4), we use a polynomial extension of the binomial coefficient $\binom{t}{q}$ from a function taking discrete values of t to a continuous function of t. By optimizing the extension of the function $g(\omega_1,\ldots,\omega_n;q)$, a value of $(\omega_1,\ldots,\omega_n)$ is obtained that can be rounded to an integral value of $\omega_1 M,\ldots,\omega_n M$ with arbitrarily high accuracy for M sufficiently large. The following proposition gives the first step towards this optimization approach.

Proposition 2: For fixed q, $g(\omega_1,\ldots,\omega_n;q)$ is a convex function of $(\omega_1,\ldots,\omega_n)$.

A proof is given in the appendix. The convexity of $g(\omega_1,\ldots,\omega_n;q)$ as a function of $(\omega_1,\ldots,\omega_n)$, as well as the existence of a concave extension as a function of q, lead to the following alternate formulation of (4).

Theorem 1: The optimization problem (4) is equivalent to

$$\begin{array}{lll} \text{maximize} & \min & g(\omega_1,\ldots,\omega_n;q) \\ q\in\{1,\ldots,q_0\} & \omega_1,\ldots,\omega_n & \\ & \text{s.t.} & \sum_{i=1}^n \omega_i = 1 \end{array} \tag{5}$$

Proof: From the proof of Proposition 1, for every ω, there exists a concave extension $\tilde{g}$ of $g(\omega_1,\ldots,\omega_n;q)$ as a function of q. Furthermore, by Proposition 2, $g(\omega_1,\ldots,\omega_n;q)$ is convex as a function of $(\omega_1,\ldots,\omega_n)$. By Minimax theorem [19], we can interchange order of max and min for the extended function $\tilde{g}$. With the linear extension of $g(\omega_1,\ldots,\omega_n;q)$, for every given ω, an optimal q is achieved at the extreme boundary point, which coincides with the solution to (4). Hence the minimum and maximum of (4) can be interchanged, yielding (5). ■

Note that the optimum of (5) need not be unique. However, each optimum gives the same value of $g(\omega_1^*,\ldots,\omega_n^*)$.

Remark 2: The interchange of min max to max min also allows us to conclude that the zero-sum game that corresponds to problems (4) and (5) admits a saddle-point point equilibrium and in case of multiple saddle points, the saddle-point strategies possess the ordered interchangeability property [19]. □

Theorem 1 leads to a straightforward bisection algorithm for computing the solution to (4). First, define $r(q)$ by

$$r(q) \triangleq \min\left\{ g(\omega_1,\ldots,\omega_n;q) : \sum_{i=1}^n \omega_i = 1 \right\}.$$

Evaluating the function $r(q)$ requires solving an equality-constrained convex program, and hence can be computed in polynomial time in $(\omega_1,\ldots,\omega_n)$. Initialize $q_{min}=1$ (representing the lower bound on q^*) and $q_{max}=q_0$ (an upper bound on q^*). At each step of the algorithm, set $q=(q_{max}+q_{min})/2$, and compute $r(q+1)-r(q)$. If $r(q+1)-r(q)>0$, then r is still increasing, and hence q_{min} is set to $q_{min}=q^*$. Otherwise, set $q_{max}=q^*$. The algorithm terminates when $q_{min}=q_{max}-1$, setting $q^*=\arg\max\{r(q_{min}),r(q_{max})\}$ and

$$(\omega_1^*,\ldots,\omega_n^*) \in \arg\min\left\{ g(\omega_1,\ldots,\omega_n;q^*) : \sum_{i=1}^n \omega_i = 1 \right\}.$$

Lemma 3: When $a_1=\cdots=a_n=a$, the global minimum of (4) occurs when $\omega_1^*=\cdots=\omega_n^*=\frac{1}{n}$.

The proof is given in the appendix. The performance of a defense strategy can be quantified by $\Gamma(\omega_1,\ldots,\omega_n)$, defined as the ratio between the cost of a non-proactive defense and a proactive defense. In a non-proactive defense, each message is authenticated using the same key, so that $\omega_1=1$ and $\omega_2=\cdots=\omega_n=0$. Hence Γ is given by

$$\Gamma(\omega_1,\ldots,\omega_n) := \frac{g^*(1,0,\ldots,0)}{g^*(\omega_1,\ldots,\omega_n)}. \tag{6}$$

The following theorem gives a lower bound on this ratio.

Theorem 2: For $M>q_0$, we have

$$\Gamma(\omega_1,\ldots,\omega_n) \geq \left[\sum_{i=1}^n \omega_i \left(\frac{\omega_i M - q_0}{M-q_0}\right)^{q_0}\right]^{-1}. \tag{7}$$

A proof appears in the appendix. Note that, when $n=1$, the bound results in $\Gamma=1$, implying that it is tight in this special case. Further, in the case of Lemma 3 where $\omega_1=\cdots=\omega_n=\frac{1}{n}$, (7) reduces to $\left(\frac{M-q_0}{M/n-q_0}\right)^{q_0}$, which increases exponentially in q_0 for $M>q_0$. Hence increasing q_0 has a greater impact on the success of the defense than increasing the number of message classes, n, and so increasing q_0 should be given higher priority for allocating computational resources when designing a proactive defense.

VI. Case Study: LQR Control

In this section, we provide a case study showing computation of $a_{i,m}, m\in\mathscr{M}_i, i=1,2,\cdots,n$, using a linear-quadratic regulator (LQR) problem as our control system model. The physical plant is described by a linear system:

$$\dot{x}(t) = Ax(t)+Bu(t), \quad x(0)=x_0, \tag{8}$$

where $x_0\in x\in\mathbb{R}^k$ is the initial condition; $x\in\mathbb{R}^k$ is the state vector; $A\in\mathbb{R}^{k\times k}$, $B\in\mathbb{R}^{k\times l}$ are constant matrices; and $u\in\mathbb{R}^l$ is the l-dimensional control input to the system. The goal of the control is to regulate the system to a given set point $\bar{x}$. Assuming that the states are perfectly observable, we design an optimal perfect-state feedback controller that minimizes the following cost criterion:

$$L(u) := \lim_{T\to\infty}\frac{1}{T}\int_0^T (\|x(t)-\bar{x}\|_Q^2 + \|u(t)\|_R^2)dt, \tag{9}$$

where $\bar{x}=\{\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_k\}\in\mathbb{R}^k$ is the setpoint configured by the system operator; $\|\cdot\|_Q^2, \|\cdot\|_R^2$ are weighted l_2-norms with symmetric square matrices $Q\in\mathbb{R}^{k\times k}, R\in\mathbb{R}^{l\times l}$ respectively, where $Q\geq 0$ and $R>0$. We restrict the optimal controller to a set of admissible perfect feedback strategies Γ, i.e., for a generic $\gamma\in\Gamma, \gamma:\mathbb{R}^k\times\mathbb{R}_+\to\mathbb{R}^l$, the control input is given by $u(t)=\gamma(x(t),t), t\geq 0$.

Theorem 3: Under the assumptions that (A,B) is controllable and (A,Q) is observable, the optimal stabilizing control is affine in the state and is given by

$$u(t)=\gamma(x(t),t) := -R^{-1}B'(Sx(t)+m), \tag{10}$$

where the control gain $S\in\mathbb{R}^{k\times k}$ is the unique solution of the algebraic Riccati equation below, within the class of positive-definite matrices:

$$A'S+SA+SBR^{-1}R'S+Q=0, \tag{11}$$

and $m\in\mathbb{R}^k$ satisfies

$$(A-BR^{-1}B'S)'m = 2Q\bar{x}. \tag{12}$$

The closed-loop system under this control is given by

$$\dot{x}(t) = (A-BR^{-1}B'S)x(t) - M\bar{x}, \tag{13}$$

where $M := 2R^{-1}B'[(A-BR^{-1}B'S)']^{-1}Q$, and the matrix $A-BR^{-1}B'S$ is Hurwitz.
This is a standard result in LQR design with non-zero set points.

We can see that the set point $\bar{x}$ affects the controller through the affine term m. Let $\hat{x}$ be the setpoint made by the malicious attacker. Denote by $\hat{K} = \{j : \bar{x}_j \neq \hat{x}_j, j = 1,2,\cdots,k\}$, which is a set of indices of setpoints compromised by the attacker. The manipulation of setpoints results in the degradation of control performances. We adopt the following metric to measure the damage.

$$J(\bar{x},\hat{x},t_0,T) = \int_{t_0}^{T} e^{-\rho(t-t_0)}\|x(t)-x_m(t)\|_Z^2 dt, \qquad (14)$$

where $\rho \in \mathbb{R}$ is the discount factor; t_0 is the time instant where the setpoint is changed, and T is the time instant when recovery strategies are applied to the system after detection, and $Z = \text{diag}\{z_1,z_2,\cdots,z_k\} \in \mathbb{R}^{k\times k}$ is a diagonal weighting matrix; $x(t), x_m(t) \in \mathbb{R}^k$ are the state trajectories generated by setpoints $\bar{x}$ and $\hat{x}$, respectively. The performance can be obtained in the closed form as follows: For a singleton set $\hat{K} = \{i\}$:

$$\begin{aligned} J(\bar{x},\hat{x},t_0,T) &= \sum_{j=1}^{k} z_j \int_{t_0}^{T} e^{-\rho(t-t_0)}(M_{ji}t)^2 dt \\ &= \left(\sum_{j=1}^{k} z_j m_{ji}^2\right)\cdot\frac{1}{\rho^3}\left((\rho^2T^2+2\rho T+2)\right. \\ &\quad \left.\cdot(-e^{\rho(t_0-T)})+(\rho^2t_0^2+2\rho t_0+2)\right) \end{aligned}$$

Let each message $m \in \mathscr{M}_i$ correspond to a setpoint change of the states of the control system indicated by the pair $(\hat{x}_{i,m}, \hat{K}_{i,m})$, where $\hat{x}_{i,m}$ is the new set point commanded by message m and $\hat{K}_{i,m}$ is the set of setpoint indices changed by message m. Depending on the content of the message, the damage $a_{i,m}$ caused by a message m can be evaluated by the physical damage as in (14), i.e., $a_{i,m} = J(\bar{x},\hat{x},t_0,T)$, where $\hat{x}$ is the setpoint value contained in message m. The parameters $a_{i,m}$ can then be used in (4) and (5) to find a saddle-point solution.

VII. Simulation

A numerical simulation study has been performed using Matlab. The simulations assume a system with $p = 8$, q_0 taking values between 2 and 10, and the number of message classes, n, taking values from 1 to 10. The control system is assumed to have $M = 100$ possible messages.

For the physical component, we consider a multivariable example given in [20], which studies the design of a controller for the lateral motion of an aircraft. The model consists of four states $x_i, i = 1,\cdots,4$, and two inputs u_1, u_2: x_1 is the bank angle, x_2 the derivative of the bank angle, x_3 is the sideslip angle, x_4 the yaw rate; u_1 the rudder deflection, and u_2 the aileron deflection. The matrices A, B in state space equation (8) are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_\beta(g/V) & N_p & N_\beta + N_\beta Y_\beta & N_r - N_\beta \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & -3.91 \\ 0.035 & 0 \\ -2.53 & 0.31 \end{bmatrix} \qquad (15)$$

We consider the following values for the aircraft parameters entering into the state matrix: $L_p = -2.93, L_\beta = -4.75, L_r = 0.78, g/V = 0.086, Y_\beta = -0.11; N_\beta = 0.1, N_p = 0.042, N_\beta = 2.601$ and $N_r = -0.29$. The matrices Q and R were set equal to the 4×4 and 2×2 identity matrices, respectively. For simplicity, we considered set points in which all states had the same value, and each message m corresponded to a different set point. The impact $a_{i,m}$ from an adversary inserting message m was computed based on (14). Each message class $\mathscr{M}_i$ consisted of messages corresponding to set points in the interval $[\chi_i, \chi_{i+1}]$, where the values of χ_i and χ_{i+1} depended on the total number of messages and the maximum state value.

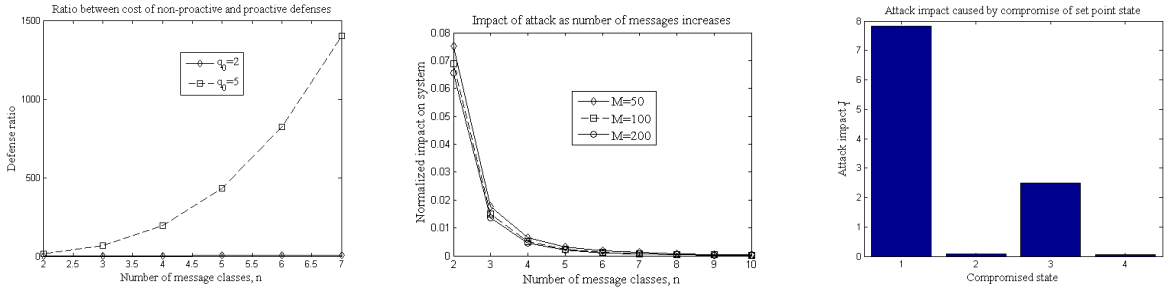
The effect of randomizing between multiple keys is illustrated in Figure 4(a). When two message classes are used, the defense ratio Γ is scaled by 50% when $q_0 = 2$ and is scaled by a factor of 14 of the non-deception case when $q_0 = 5$. As the number of message classes increases, the ratio grows exponentially, agreeing with Theorem 2. Furthermore, the rate of increase is exponentially larger as the parameter q_0 increases.

The impact of the total number of messages on the adversary's success probability is shown in Figure 4(b). Increasing the number of messages reduces the expected impact of the attack on the system, as the adversary has a reduced probability of finding a set of messages belonging to the same class. Increasing the number of messages from 100 to 200, however, provided no improvement in security.

Figure 4(c) shows the impact on the system, J_i, caused by an adversary introducing a malicious set-point for state x_i, for $i = 1,\ldots,4$. Each message class $\mathscr{M}_i$, $i = 1,\ldots,4$, corresponds to a different state variable, with each representing a set point for that state. Altering the set-point of state 1 has a significantly larger effect on the system than altering states 2 or 4, implying that messages that alter state 1 should be assigned to a message class with fewer elements. This will result in a lower probability that a message modifying the set-point of state 1 can be injected. Conversely, changing the set-point of state 2 has lower impact, and hence messages altering state 2 should belong to a set $\mathscr{M}_j$ with a higher value of M_j.

VIII. Conclusion

In this paper, we have studied the problem of mitigating attacks on control systems, such as Stuxnet-type malware, using proactive mechanisms. We have introduced a novel proactive defense, in which the system randomizes between



(a) (b) (c)

Fig. 4. Numerical evaluation of proposed deception mechanism. (a) The deception ratio, defined as the ratio of the expected impact of the attack when the number of message classes is n and the impact when there is only one message class. The adversary's success probability decreases exponentially as the number of message classes increases, with the rate of decrease determined by the parameter q_0. (b) The effect of the total number of messages M on the impact of the attack. Increasing the message space makes it more difficult for the adversary to forge messages. (c) The impact of compromising each of the four states. Compromise of state 1 results in the largest deviation from the desired set point, resulting in a higher a_i value.

different cryptographic keys for authentication. Based on security metrics such as the adversary's probability of success and the impact of a successful attack, we have developed an analytical framework for selecting the number of messages signed using each key. The interactions between the network and the adversary can be viewed as a zero-sum game. We provide an efficient algorithm for finding a saddle-point equilibrium of the game, leading to optimal worst-case selection of the number of messages of each type. Worst-case bounds have been proven, showing that the proposed scheme can achieve arbitrarily low adversary success probability for a sufficiently large number of keys. As a case study, we have considered a cyber-physical system with a linear quadratic regulator similar to the frequency drives targeted by Stuxnet through both theoretical analysis and simulation study.

REFERENCES

[1] Institute for Science and International Security, "Did Stuxnet take out 1,000 centrifuges at the Natanz enrichment plant?" December 2010.

[2] E. Byre, A. Ginter, and J. Langill, "How Stuxnet spreads – a study of infection paths inbest practice systems," *Tofino Security White Papers*, February 2011.

[3] N. Falliere, L. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[4] D. Beresford, "Exploiting Siemens Simatic S7 PLCs," *Black Hat USA*, 2011.

[5] D. Jin, D. Nicol, and G. Yan, "An event buffer flooding attack in dnp3 controlled scada systems," *Proc. of the 2011 Winter Simulation Conference (WSC)*, pp. 2614–2626, 2011.

[6] R. Bobba, H. Khurana, M. AlTurki, and F. Ashraf, "PBES: a policy based encryption system with application to data sharing in the power grid," pp. 262–275, 2009.

[7] D. Stinson, *Cryptography: Theory and Practice*. CRC press, 2006.

[8] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," *Proc. of the Twenty-first Annual ACM Symposium on Theory of Computing*, pp. 33–43, 1989.

[9] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special Publication*, vol. 800, p. 82, 2007.

[10] A. Teixeira, S. Amin, H. Sandberg, K. Johansson, and S. Sastry, "Cyber security analysis of state estimators in electric power systems," *Proc. of 49th IEEE Conference on Decision and Control (CDC)*, pp. 5991–5998, 2010.

[11] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "SCADA security in the light of cyber-warfare," *Computers & Security*, pp. 418–436, 2012.

[12] "Eugene kaspersky unveils plans for new secure SCADA OS," http://threatpost.com/en_us/blogs/eugene-kaspersky-unveils-plans-new-secure-scada-os-101612.

[13] H. Shacham, M. Page, B. Pfaff, E. Goh, N. Modadugu, and D. Boneh, "On the effectiveness of address-space randomization," *Proc. of the 11th ACM Conference on Computer and Communications Security*, pp. 298–307, 2004.

[14] V. Pappas, M. Polychronakis, and A. D. Keromytis, "Smashing the gadgets: Hindering return-oriented programming using in-place code randomization," *Proc. of the 2012 IEEE Symposium on Security and Privacy*, pp. 601–615.

[15] Q. Zhu and T. Başar, "Towards a unifying security framework for cyber-physical systems," in *Proc. of the 2nd Workshop on Foundations of Dependable and Secure Cyber-Physical Systems*, 2011.

[16] ——, "Robust and resilient control design for cyber-physical systems with an application to power systems," *Proc. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 4066–4071, 2011.

[17] Q. Zhu, C. Rieger, and T. Başar, "A hierarchical security architecture for cyber-physical systems," in *Proc. of 4th International Symposium on Resilient Control Systems*, August 2011.

[18] US-CERT, "Improving industrial control systems cybersecurity with defense-in-depth strategies," October 2009.

[19] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. SIAM Series in Classics in Applied Mathematics, 1999.

[20] B. Polyak and R. Tempo, "Probabilistic robust design with linear quadratic regulators," *Systems & Control Letters*, vol. 43, no. 5, pp. 343–353, 2001.

APPENDIX

Proofs of Lemmas 2 and 3, Propositions 1 and 2, and Theorem 2 are given as follows.

Proof of Lemma 2: Suppose $\mathbf{Adv} \in \mathscr{A}$, so that $\mathbf{Adv}$ calls $\mathbf{Adv}_0$ as a subroutine with parameter p and number of queries q. Let $\hat{M}$ denote the number of (message, MAC) pairs passed as input to $\mathbf{Adv}_0$ with messages in $\mathscr{M}_1$. Then, the success probability $P_s(p,q,M;\mathbf{Adv})$ is given by

$$P_s(p,q;\mathbf{Adv}) = \sum_{m=1}^{q} \mathbf{Pr}(\text{success } |\hat{M}=m)\mathbf{Pr}(\hat{M}=m). \quad (16)$$

Now, Assumption 2 of Section IV implies that

$$\mathbf{Pr}(\text{success } |\hat{M}=m) = \begin{cases} f(p,q), & m \in \{0,q\} \\ 0, & \text{else} \end{cases} \quad (17)$$

Furthermore, Assumption 1 implies that the adversary cannot differentiate between messages in $\mathscr{M}_i$ and $\mathscr{M}_j$, so that the messages input to $\mathbf{Adv}_0$ are chosen uniformly at random. This leads to

$$\mathbf{Pr}(\hat{M}=m) = \frac{\binom{M_1}{m}\binom{M_2}{q-m}}{\binom{M}{q}}. \quad (18)$$

Combining these equations leads to

$$P_s(p,q;\mathbf{Adv}) = \frac{\binom{M_1}{q}}{\binom{M}{q}} f(p,q) + \frac{\binom{M_2}{q}}{\binom{M}{q}} f(p,q). \tag{19}$$

Since this expression depends only on q, maximizing over **Adv** yields (3). ∎

Proof of Proposition 1: First, since the log function is increasing, the function g is nondecreasing (resp. nonincreasing) on that interval if $\log g$ is nonincreasing (resp. nondecreasing on an interval). In what follows, we show that there exists a concave extension to $\log g(\omega_1,\ldots,\omega_n)$, and hence a unique maximum point q^* satisfying the criteria of the proposition. For $q \in \{1,\ldots,q_0\}$, the log function of g is given by

$$\log g(\omega_1,\ldots,\omega_n;q) = \log\left(\sum_{i=1}^n a_i^* \frac{\binom{M_i}{q}}{\binom{M}{q}} \left(\frac{1}{p}\right)^{q_0-q}\right).$$

It suffices to find a concave extension of

$$q\log p + \log\left[\sum_{i=1}^n a_i^* \frac{\binom{M_i}{q}}{\binom{M}{q}}\right]. \tag{20}$$

Furthermore, since the first term has a linear extension, it remains to find a concave extension of the second term of (20). To do so, we find a concave extension $\tilde{\alpha}(q)$ of the function

$$\alpha(q) = \sum_{i=1}^n a_i^* \frac{\binom{M_i}{q}}{\binom{M}{q}}$$

and then choose $\log\tilde{\alpha}(q)$ as the concave extension of the second term. To compute $\tilde{\alpha}(q)$, the goal is to show that, for $1 \le q \le' \le q_0$,

$$\alpha(q+1) - \alpha(q) \ge \alpha(q'+1) - \alpha(q'), \tag{21}$$

which allows a piecewise linear concave extension with slope $h(q+1)-h(q)$ on the interval $[q,q+1]$. By definition of $\binom{M_i}{q}$, (21) is equivalent to

$$\sum_{i=1}^n a_i^* \prod_{j=0}^{q} \frac{M_i-j}{M-j} - \sum_{i=1}^n a_i^* \prod_{j=0}^{q-1} \frac{M_i-j}{M-j} \ge \sum_{i=1}^n a_i^* \prod_{j=0}^{q'} \frac{M_i-j}{M-j} - \sum_{i=1}^n a_i^* \prod_{j=0}^{q'-1} \frac{M_i-j}{M-j},$$

which in turn reduces to

$$\sum_{i=1}^n \left[a_i^* \left(\prod_{j=0}^{q-1} \frac{M_i-j}{M-j}\right)\left(\frac{M_i-q}{M-q}-1\right)\right] \ge \sum_{i=1}^n \left[a_i^* \left(\prod_{j=0}^{q'-1} \frac{M_i-j}{M-j}\right)\left(\frac{M_i-q'}{M-q'}-1\right)\right]. \tag{22}$$

Now, since $M_i \le M$, $M_i - j \le M$, and so

$$\prod_{j=0}^{q-1} \frac{M_i-j}{M-j} \ge \prod_{j=0}^{q'-1} \frac{M_i-j}{M-j}$$

for $q \le q'$. Similarly, $\frac{M_i-q}{M-q} \ge \frac{M_i-q'}{M-q'}$. These two identities imply (22), which then implies that α, and hence $\log\alpha$, have concave extensions. Thus the function (20) has a concave extension, which has a unique maximum point $\tilde{q}$ satisfying the conditions of the proposition. Setting

$$q^* = \arg\max_q \{g(\omega_1,\ldots,\omega_n;\lfloor\tilde{q}\rfloor), g(\omega_1,\ldots,\omega_n;\lceil\tilde{q}\rceil)\}$$

yields the desired result. ∎

Proof of Proposition 2: We have that $\frac{d}{dt}\binom{t}{q} = \binom{t}{q}\sum_{j=0}^{q-1}\frac{1}{t-j}$, so that

$$\frac{d}{d\omega_i}\binom{\omega_i M}{q} = M\binom{\omega_i M}{q}\sum_{j=0}^{q-1}\frac{1}{\omega_i M - j}.$$

Differentiating again with respect to ω_i yields

$$\frac{d^2}{d\omega_i^2}\binom{\omega_i M}{q} = M^2\binom{\omega_i M}{q}\left[\left(\sum_{j=0}^{q-1}\frac{1}{\omega_i M-j}\right)^2 - \sum_{j=0}^{q-1}\frac{1}{(\omega_i M-j)^2}\right] \ge 0. \tag{23}$$

Eq. (23) implies that $g(\omega_1,\ldots,\omega_n;q)$ is a nonnegative weighted sum of convex functions, and is therefore convex. ∎

Proof of Lemma 3: For a fixed q, the inner optimization problem of (5) is written using a Lagrange multiplier λ as follows:

$$\sum_{i=1}^n a\frac{\binom{\omega_i M}{q}}{\binom{M}{q}} f(p,q)\omega_i + \lambda\left(\sum_{i=1}^n \omega_i - 1\right).$$

Differentiating with respect to ω_i yields

$$\frac{a}{\binom{M}{q}} f(p,q)\left[\sum_{j=0}^{q-1}\left(\frac{\omega_i M}{\omega_i M-j} + \binom{\omega_i M}{q}\right)\right] + \lambda = 0. \tag{24}$$

Choosing $\omega_1 = \ldots = \omega_n = \frac{1}{n}$ satisfies (24) for $i = 1,\ldots,n$, as well as primal feasibility, and hence is optimal. ∎

Proof of Theorem 2: Let a^* denote the impact of the most damaging forgery. Then Γ can be expressed as

$$\Gamma(\omega_1,\ldots,\omega_n) = \frac{a^* f(p,M)}{\max_q \sum_{i=1}^n a_i^* \frac{\binom{\omega_i M}{q}}{\binom{M}{q}} f(p,q)\omega_i},$$

leading to the bound

$$\begin{aligned}\Gamma(\omega_1,\ldots,\omega_n) &\ge \frac{f(p,M)}{\sum_{i=1}^n \max_q \frac{\binom{\omega_i M}{q}}{\binom{M}{q}} f(p,q)\omega_i} \\ &\ge \frac{f(p,M)}{\sum_{i=1}^n \max_q \left(\frac{\omega_i M-q}{M-q}\right)^q f(p,q)\omega_i} \\ &= \frac{f(p,M)}{\sum_{i=1}^n \max_q \left(\frac{\omega_i M-q}{M-q}\right)^q \left(\frac{1}{p}\right)^{(q_0-q)_+}\omega_i}.\end{aligned}$$

For $M > q_0$, $f(p,M) = 1$, and the denominator is maximized by $q^* = \min\left\{\frac{M(p-\omega_i)}{p-1}, q_0\right\} = q_0$. Substitution yields the desired result. ∎