ever, exhibit interdependencies between potential targets of attack. These may be explicit, as in IT and supply chain network security, or implicit, as in defending critical infrastructure (where, for example, successful delivery of transportation services depends on a highly functional energy sector, and vice versa), or in securing complex software systems (with failures at some modules having potential to adversely affect other modules). While in such settings the assumption of independence seems superficially violated, we demonstrate below that under realistic assumptions about the nature of interdependencies, we can nevertheless leverage the highly scalable optimization techniques which assume independence.

In all, we offer the following contributions. (1) We modify and extend the previous linear programming techniques for Stackelberg security games to allow for an arbitrary set of security configurations (rather than merely to cover a target, or not), as well as to account for both random and targeted failures, and replace hard constraints on defense resources with costs associated with specific security configurations (Section 3). (2) We present and justify a crucial assumption on the nature of interdependencies that allows us to use our LP formulations which fundamentally assume independence between targets (Section 4.1). We then offer a simple model of interdependencies based on probabilistic failure cascades satisfying this assumption. Our model makes an explicit distinction between an intrinsic and indirect value of assets, the latter being due entirely to interdependencies. This allows an economically meaningful extension of a well-known independent cascade approach to modeling the spread of infectious diseases or ideas (Section 4). (3) We demonstrate that for trees we can compute expected utilities for all targets in linear time (Section 4). (4) We extend our model to capture uncertainty about network structure, and experimentally study the impact of such uncertainty (Sections 4.5 and 6.1). (5) We show that our approach is both scalable to realistic security settings and offers much better solutions than state-of-the-art alternatives (Sections 5.2 and 5.2). (6) We experimentally study the properties of optimal defense configurations in real and generated networks (Section 6).

2 Stackelberg Security Games

A Stackelberg security game consists of two players, the leader (defender) and the follower (attacker), and a set of possible targets. The leader can decide upon a randomized policy of defending the targets, possibly with limited defense resources. The follower (attacker) is assumed to observe the randomized policy of the leader, but not the realized defense actions. Upon observing the leader's strategy, the follower chooses a target so as to maximize its expected utility.

In past work, Stackelberg security game formulations focused on defense policies that were costless, but resource bounded. Specifically, it had been assumed that the defender has $K$ fixed resources available with which to cover targets. Additionally, security decisions amounted to covering a set of targets, or not. While in numerous settings to which such work has been applied (e.g., airport security, federal air marshal scheduling) this formulation is very reasonable, in other settings one may choose among many security configurations for each valued asset, and, additionally, security resources are only available at some cost. For example, in cybersecurity, protecting computing nodes could involve configuring anti-virus and/or firewall settings, with stronger settings carrying a benefit of better protection, but at a cost of added inconvenience, lost productivity, as well as possible licensing costs. Indeed, costs on resources may usefully replace resource constraints, since such constraints are often not hard, but rather channel an implicit cost of adding further resources.

While security games as described above naturally entail an attacker, most systems exhibit failures that are not at all a deliberate act of sabotage, but are due entirely to inadvertent errors. Even though such failures are generally far more common than attacks, the vast majority of work in security games posits an attacker, but ignores such failures entirely; essentially the lone exception is a paper by Zhuang and Bier [2007] which offers an analytic treatment of a simple model making explicit the distinction between attacks and natural disasters. Our formulation below is, to our knowledge, the first to explicitly model both attacks and random failures in the Stackelberg security game literature.

To formalize, suppose that the defender can choose from a finite set $O$ of security configurations for each target $t \in T$, with $|T| = n$. A configuration $o \in O$ for target $t \in T$ incurs a cost $c_{o,t}$ to the defender. If the attacker happens to attack $t$ while configuration $o$ is in place, the expected value to the defender is denoted by $U_{o,t}$, while the attacker’s value is $V_{o,t}$. A key assumption in Stackelberg security games is that the targets are completely independent: that is, player utilities only depend on the target attacked and its security configuration [Kiekintveld et al., 2009]. We revisit this assumption below when we turn to networked (interdependent) settings. We denote by $q_{o,t}$ the probability that the defender chooses $o$ at target $t$. Finally, let $r$ be the prior probability of the defender that a failure will happen due to a deliberate attack. If no attack is involved, any target can fail; the defender’s belief that target $t$ randomly fails (conditional on the event that no attack is involved) is $g_t$, with $\sum_t g_t = 1$. 