Diversity, Safety and Security in Embedded Systems: modelling adversary effort and supply chain risks

Ilir Gashi, Andrey Povyakalo, Lorenzo Strigini
Centre for Software Reliability, City University London, London, U.K.
{ I.Gashi, A.A.Povyakalo, L.Strigini}@city.ac.uk

***Abstract*— We present quantitative considerations for the design of redundancy and diversity in embedded systems with security requirements. The potential for malicious activity against these systems have complicated requirements and design choices. New design trade-offs have arisen besides those already familiar in this area: for instance, adding redundancy may increase the attack surface of a system and thus increase overall risk. Our case study concerns protecting redundant communications between a control system and its controlled physical system. We study the effects of using: (i) different encryption keys on replicated channels, and (ii) diverse encryption schemes and implementations. We consider two attack scenarios, with adversaries having access to (i) ways of reducing the search space in attacks using random searches for keys; or (ii) hidden major flaws in some crypto algorithm or implementation. Trade-offs between the requirements of integrity and confidentiality are found, but not in all cases. Simple models give useful design insights. In this system, we find that key diversity improves integrity without impairing confidentiality – no trade-offs arise between the two – and it can substantially increase adversary effort, but it will not remedy substantial weaknesses of the crypto system. Implementation diversity does involve design trade-offs between integrity and confidentiality, which we analyse, but turns out to be generally desirable for highly critical applications of the control system considered.**

Keywords— *security assessment; safety assessment, safety vs. security trade-offs; adversary effort; embedded systems; software and hardware diversity*

I. INTRODUCTION

Security in embedded system is an important concern, highlighted by successful attacks on safety-critical systems and national infrastructures and by widely voiced concerns about widespread vulnerabilities. We address the design decisions and trade-offs arising when using diverse, redundant components to address safety and security concerns.

In critical embedded systems, it is common to apply, for reliability and for safety [1], [2]:

- *redundancy*: using more than one functional modules (achieving the same system goals), to improve the likelihood that the function is performed correctly, or avoids unsafe failures;
- *diversity*: intentional differences between redundant components, to reduce the likelihood of common failures due to systematic causes that would reduce the benefit of redundancy.

Redundancy and diversity are also useful for security, e.g. via redundant defences (*viz.* diverse antivirus software or firewalls), or redundant assets (*viz.* multiple diverse servers for availability); a designer has to anticipate their effects with respect to all faults, accidental or malicious, and any design trade-offs arising.

In addressing security concerns, we have often heard objections to probabilistic methods, on the basis that security deals with unpredictable, *intentional* human action. The argument *for* using probabilities ([3], [4]) can be summarized as: quantitative probabilistic reasoning is a means for reasoning rationally in the presence of uncertainty, although there is no claim that it eliminates it. This paper is an example of this use.

This paper addresses design decisions about the redundancy and degree of diversity in a redundant architecture, in view of the possible need of trade-offs between competing requirements. We use a simple example, generalizing from a concrete industrial case study [5]: a controller for an electro-mechanical system, implemented as three parallel channels with 2-out-of-3 voting: if any one out of three channels fails, the others can continue delivering correct control, or trigger a transition to a fail-safe state (what is "fail-safe' depends on the controlled mechanical load).

For a safety requirement like "no hazardous condition shall be caused by this controller", the chances of it being satisfied during operation depend on the probability of two or more channels failing together so as to cause dangerous control inputs, due to any combination of malicious or accidental causes.

In this kind of systems, the security concern is usually "security for safety": an adversary may produce an accident, or make it more likely and the designers' concern is to make this less likely. In security terminology, this creates *integrity* requirements (we want the adversary not to be able to cause these failures). Through violations of integrity, an adversary may cause accidents: directly, at a time of his choosing, by making a majority of channels agree on an unsafe action; or indirectly, by making one or more of the channels unable to react properly when another channel fails accidentally, making the system effectively non-redundant and thus less safe than is required: a delayed-effect, "stealthy" attack. Adversaries may also be interested in less severe forms of sabotage, to reduce the availability or efficiency of the system, e.g. perturbing the control algorithms so as to increase energy consumption, wear-and-tear, stress on operators/users, etc.

So far, introducing security considerations has not changed the characterization of this system as a 2-out-of-3 system: it only fails if two or more channels fail, for any reason. The adversary's goal for an integrity violation is to emulate carefully chosen failures for two out of three channels. However, any embedded system may also have *confidentiality* requirements. That is, the three channels process some information such that it becoming known to an adversary would be a loss, despite not having direct effect on data/system integrity. Confidentiality requirements may arise for instance:

- independently of the safety requirements for the individual system compromised, e.g. being aimed at safeguarding intellectual property in a channel's software, or in the controlled system (e.g., sensor data may reveal design details);
- or from safety concerns, indirectly: e.g., by reading I/O data, the adversary might devise better attacks ("cyber" attacks or physical attacks), on the same or even on different applications of the same control system. In other systems, attackers may gain access to code which will, similarly, facilitate other attacks.

Importantly, if the confidential information can be obtained through any channel, then our example system behaves – from the confidentiality viewpoint – as a "series" or "3 out of 3" system: compromising one channel compromises the whole. Adding more redundant channels (identical or diverse) would decrease risk due to accidental faults, but typically increase risk of violation of confidentiality. Trade-offs would arise between confidentiality and safety and between direct and indirect safety risk. The optimum degree of redundancy depends on the combination of the adversary's strategy and the particular loss function that associates losses to the various loss events.

The example that motivates this study arose in project SeSaMo (http://sesamo-project.eu/), [5] which studied synergies and trade-offs between security and safety in embedded systems. The problem is how to protect redundant communication channels, in an embedded system, between a controlled apparatus and its feed-back controller. Redundant communication channels bring sensor readings to the controller and bring control inputs back from it. Attackers might gain access (read and/or write) to these communication channels; so the messages are encrypted to prevent the attacker discovering their contents or injecting forged messages (forged control commands, or forged sensor readings, causing the controller to issue inappropriate commands), so as to cause harm to the controlled system.

The rest of the paper is structured as follows: section II introduces our case study; section III describes the attack scenarios and modelling assumptions; section IV analyses our triple-modular redundant system under a cryptanalysis attack, in which the adversary systematically explores a search space for the key used in each channel; section V describes a similar analysis for the case in which some of the crypto components used in the implementation contain serious flaws known to the adversary (we call this scenario "supply chain" attack); section VI discusses our results in the light of other research work; section VII presents conclusions and discusses generalizations and further work.

II. Case study

Our case study is a *motion control system*: an electric motor, driven by solid-state inverters, and a *Controller* which controls the latter to achieve the motion required (continuous, variable speed, rotations of a mechanical load from a position to another, etc).

The angular position of the motor is sensed by triple sensors and their readings sent to the Controller, which uses them to calculate the control inputs for the power electronic components which drive the motor. All signals are replicated for reliability and encrypted for security. Fig 1 shows this for the direction from the motor to the controller.

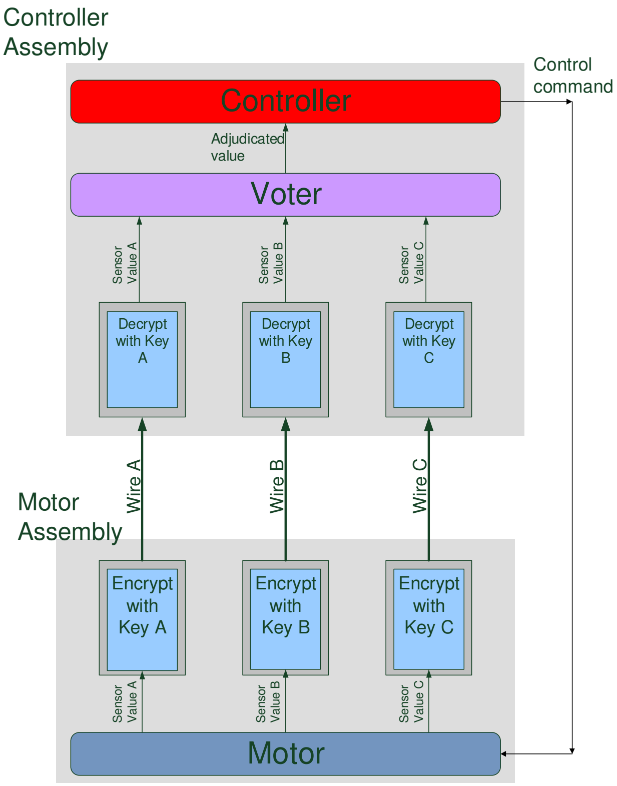


Fig. 1. A motion control system: a high level overview of the components in the Motor Assembly and the Controller Assembly. We only show the triplication of the sensor values and encryption/decryption followed by voting before an adjudicated value goes to the controller (i.e. upwards in the figure above from Motor Assembly to Controller Assembly). Triplication of control messages, encryption/decryption and voting also happens with the "control command" that comes out of the controller and goes to the motor assembly (i.e. downwards in the figure) – we omit it from the figure for the sake of readability. Other components like the Voter and the Controller can also be replicated for reliability, but this does not affect security in our attack scenario: therefore it is not shown in the figure.

For our purposes, this system is composed of these subsystems:

1. the Motor Assembly containing the motor and power electronics; the components for receiving, decrypting and voting the control inputs; the sensors, and the components (including encryption components) for transmitting their

readings. A hardware-generated synchronization signal triggers periodic transmission of sensor readings;

2. the Controller Assembly with components performing: decryption of sensor readings; voting on them, to give a single sensor input to the controller; the control function proper, which calculates control signals; the encryption and transmission of the latter. The controller assembly is a standard subsystem that can be connected to a variety of motors with their mechanical loads;
3. the communication channels between the two subsystems above.

Our problem is the attack interface provided by the communication media. The Motor assembly and Controller assembly are positioned at some distance from each other, so that adversaries are interested in intercepting and/or forging messages in the communication channel between the two. In our example this could be several meters of communication cables, on which the adversary could briefly install taps to intercept messages and again at a later time to inject forged messages. In other embedded systems the communication channels could be e.g. buses or wireless channels, each varying the opportunities and problems for the attackers in reading or injecting messages.

The encryption on the messages is thus the obstacle that the attacker has to overcome. Encryption is organised as follows (we use sensor reading messages in channel A as an example):

- from the motor assembly unit, sensor A's readings are to be sent to the controller assembly through a dedicated communication channel (wire A), after encryption;
- the Sensor A reading and Sensor A ID together with protocol information (such as a sequence number and checksum) form a message; this is encrypted via a symmetric-key algorithm with key K_A to produce an encrypted message;
- the encrypted message is sent over wire A to the Controller Assembly;
- when received by the Controller Assembly, the message is decrypted using the same symmetric key K_A. Authenticity is checked; design options for this include to consider the message authentic if the decryption produces a legal plaintext (recognizable by e.g. including in the plaintext an error detection code); or to also send a cryptographic hash for authentication. For the sake of brevity, we only analyse the former option: if decryption succeeds (it reveals a correct checksum), then the message is considered authenticated;
- the three keys K_A, K_B and K_C may be different. In principle, the system designer may decide to use for the three channels the same key or different keys; and to use the same or different encryption algorithms. For simplicity, we will assume in either case the *same key length* for all channels.

This system exhibits some security concerns present in many embedded systems:

- the electronics are intended to be long-lived: both the encryption algorithms and their keys are hardwired and will not change over the many years of operation of an installed system. Thus:
    - a patient adversary can intercept a sequence of messages from a system controlling a high-value item of machinery and then spend as long as needed (perhaps months, perhaps five or ten years) performing offline cryptanalysis, to discover keys so as to perform an attack;
    - in particular, the longer an encryption algorithm (or a specific implementation) is in use, the more likely it is that vulnerabilities/weaknesses will be found/published for it: security risk may increase dramatically over a system's lifetime. With increasing openness of embedded systems to a dynamic environment, there are pressures to make embedded systems easy to update/patch, but this also brings drawbacks so that purely static embedded systems are, for the time being, an important scenario for security analysis;
- additionally, keeping design details secret is not a viable option. Message formats will be in the documentation of the off-the-shelf components that compose the system. The plaintext of the message will for some installations have known ranges of possible values (e.g., the angle of rotation of a given machine driven by an instance of this system). For design details that cannot be obtained from published documentation, attackers may have the option of buying an instance of the control system to study in their own lab.

III. Attack Scenarios, Means and Goals; Modelling Assumptions

We consider two possible means of attacks on security of the communication channels:

- **"Cryptanalysis"** by search of a reduced key space. We assume here the simplest form of cryptanalysis: random search of a key space, so that the probability of finding the key increases linearly with the effort. Current symmetric encryption methods with sufficient key length, like AES, are considered unbreakable by brute force in feasible amounts of time. But a reasonable concern, given the history of cryptography, is that an adversary may become able to substantially reduce the key space to search, e.g. by knowing the algorithm by which the vendor chooses the encryption keys, or other implementation flaws[1]. In this scenario, the adversary intercepts some messages containing sensor readings and attempts to discover the encryption keys used. To this end he will exploit flaws in physical security, e.g. access to the area during normal operation to install and remove taps, corrupt maintenance staff to record messages

[1] Modelling this scenario also gives us a general model for all attacks that rely on systematic search of a space for a "winning" element, with the same probability for all elements.

while performing diagnostics, etc. Bus-based or wireless communication may offer other opportunities. The adversary needs to listen on communication lines just long enough to collect messages to use in the cryptanalysis effort; in the limiting case just one message.

- "**Supply chain**" attacks: exploiting fatal vulnerabilities in the encryption, that allow very low-effort attacks, and are known to the attackers but not to the system designers. A common concern is a "corrupted" supply chain, through which vulnerabilities were intentionally inserted for later exploitation. Some scenarios are: an intentional flaw in the implementation of the encryption; keys extracted from a reduced key space, so that an adversary who knows this restricted set can apply random search attacks with a chance of success at affordable cost; or, the key associated with a specific instance of a component is directly leaked to the adversary. Similar effects would happen if *unintended* vulnerabilities, poor choice of keys, secret lists of keys are discovered by, or sold to, the adversary.

We consider two possible goals for attacks:

- "confidentiality" breach: discovering the key for one communication channel, so as to decipher the sequence of messages in it;
- "integrity" breach: discovering the keys for a majority of channels (2 out of 3, in our case) so gaining the ability to forge messages that will cause erroneous and possibly dangerous control signals at the output of the voter in the Motor Assembly; or that will cause erroneous sensor readings at the outputs of the voter in the Controller Assembly, so that these false readings deceive the controller into issuing dangerous control signals.

We do not consider an adversary who only aims at causing unavailability by jamming the communication channels, which the assumed ability for physical access makes easy, but would be easy to detect and counteract. Likewise, we do not consider attacks in which the adversary, having breached security on only one channel, injects erroneous messages in that channel, as a "stealth" attack on integrity that would cause a majority of erroneous signals the next time that one of the non-compromised channels fails. Again, such an attack would likely be quickly detected by usual sanity checks on message values, and thus fail by triggering a maintenance intervention and the discovery of the intrusion.

Once an adversary acquires ways of injecting forged messages, whether and when he will exploit this capability to cause harm depends on his intentions and circumstances.

The parameter of interest for the system designer will be the *probability of the attacker having or acquiring* this capability if he pursues it. We will thus study this probability. This is consistent with the common practice in safety analysis of assessing and containing the probability of hazard conditions rather than that of accidents. The time at which an adversary may decide to pursue these capabilities, or, after acquiring them, to strike, will instead be of interest during operation and a matter of intelligence about the attackers' intentions, resources and strategies.

We study the security effects of design choices as a function of model parameters, as though these parameters are known, thus leaving the problem of parameter estimation as a separate problem. This divide-and-conquer approach gives useful insight to designers (as experience with design for reliability and safety shows), though to yield design decision it also requires some assumptions about, or estimation of, model parameters.

IV. CRYPTANALYSIS ATTACKS VIA RANDOM SEARCH, AND DEFENCE BY KEY DIVERSITY

We study the amount of "adversary effort" required, an analogue of the "time to failure" in reliability. One can translate "adversary effort" into time via assumptions about the intentions, resources and capabilities of the adversaries; but this translation is often unnecessary for the purpose of just discovering design trade-offs and even for optimizing design. We are interested in how redundancy or diversity change the effort needed for the adversary to achieve a certain probability of success. Our measure of effort will be the number t of attempts, that is, of possible key values that the attacker tries out.

Increasing this required effort is the designer's goal, both for reducing the probability of an attacker succeeding and for deterring attacks, or causing an attacker to give up and move to a different target. Even in these days of high-performance distributed cryptanalysis with cheap computing resources, an adversary needs to decide whether spending, or continuing to spend, resources against a certain target is promising enough to be a wise choice, in comparison with alternative uses of the resources.

We call T_{exh} the number of attempts required for an exhaustive search (i.e. the size of the keyspace subset to search), and we consider the probability of an attacker achieving a goal (discovery of one, or two, or three channel keys) as a function of the number of attempts t. Assuming that the keys are allocated to channels by choosing randomly and independently from the whole keyspace (as is reasonable)[2], the event "the r-th key tried by the attacker is the right key for channel i" is independent of any event of the same form affecting other channels. This set of independence properties underlies the following results. Last, for notational convenience we define a measure of "normalized effort", $\tau = \frac{t}{T_{exh}}$.

The adversary may choose to attempt decryption of one, two or three channels *in parallel* (that is, spreading trials of possible keys - the same or different ones - on messages from the three channels) or *in sequentially* (dedicating all the effort to one channel first, and only after success on that one moving on, if desired, to the next channel).

[2] An alternative assumption is that the only probabilistic dependence between keys assigned to the channels is that they must be different. The effect on the results presented here is numerically negligible, unless the keyspace is minuscule.

The sequential attack is the more efficient process for the attacker[3] (see Appendix). With this form of attack:

- as a baseline, we observe that with just one channel, the probability of success is $Q_1 = \frac{t}{T_{exh}} = \tau$;
- with three channels sharing a single key the probability of success (giving the ability to read and/or forge messages on any number of channels) is still $Q_1 = \frac{t}{T_{exh}} = \tau$;
- with three different keys, the probability of finding one key is still $Q_1 = \frac{t}{T_{exh}} = \tau$ (since it turns out that the best attack strategy is to concentrate effort on one channel). *Redundancy and diversity do* ***not*** *increase the risk to confidentiality*[4];
- from the viewpoint of integrity (breaking two out of the three keys), the expression Q_2 for the probability of success is more complex but is plotted in Fig 2. For *t* such that $1<<t< T_{exh}$, $Q_2 = \frac{t(t-1)}{2\,T_{exh^2}} \approx \frac{1}{2}(\tau)^2$: using diverse keys reduces the adversary's probability of success by a factor $\frac{\tau}{\frac{1}{2}\tau^2} = \frac{2}{\tau}$;
- in other words, it reduces this probability from a value Q_1 to $Q_2 = \frac{(Q_1)^2}{2}$.

If, instead, the adversary chose the "parallel", sub-optimal strategy of attack, his best option would be to split the effort equally over the three channels, giving probability of success $\frac{1}{3}\tau^2 - \frac{2}{27}\tau^3 \approx \frac{1}{3}\tau^2$. Therefore the probability of success is lower by a 1/3.

An alternative description is in terms of adversary effort required to achieve a certain probability of success. The effort required for *assured* decryption of *m* keys is of course $m\, T_{exh}$, so, using diverse keys doubles the adversary's effort for an integrity breach on the 2-out-of-3 system (that is, for finding two keys). However, an adversary may well consider worthwhile an attack that gives a non-negligible probability of success, say 10% or 20%; or even much less. Fig 2 shows that the lower the probability of success that the adversary considers sufficient to justify attacks (which often also means "the higher the value of the target", making these moderate-probability attacks important for the defender – e.g. leading to severe, though low-probability, accidents), the more advantage diversity of keys will give to the defence.

A useful measure of this advantage of using different keys is the increase in the adversary effort required to achieve a certain probability of success. Fig 3 gives an example[5].

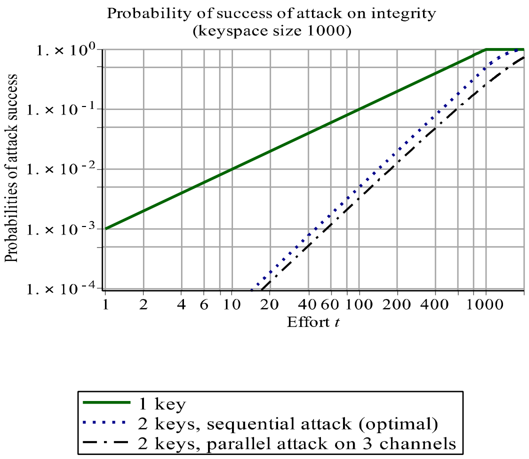


Fig. 2. Probability of success for a given effort, for finding one key, for finding two via the optimal strategy, and for finding two via one of the non-optimal strategies (dividing effort equally on the three channels) that an adversary may follow.

If we believe our adversary to consider worthwhile an attack that gives a probability of success of – say – 10%, then, using the fact that $Q_1 = \frac{t}{T_{exh}} = \tau$ and the approximation $Q_2 \approx \frac{1}{2}(\tau)^2$, we can observe that a probability of success that with one key is given by effort τ requires, with diverse keys, effort $\sqrt{2\,\tau}$: larger than τ for any search short of a complete search of both key spaces. To consider which conditions favour the defender over the attacker, we can rewrite this expression without the normalization: the level of Q_1 achieved in *t* attempts with a one-key system requires, with two keys, $\sqrt{2\,T_{exh} t}$. The ratio of the effort required for two keys vs one key is $\sqrt{\frac{2}{\tau}}$: the smaller the subset of the key space that the attacker can afford to search, the greater (by an *inverse square-root* law) the effort increase required by the presence of diverse keys.

Considering an attacker's options, we see that within a wide range of values, the smaller the probability of success that the adversary considers worth an attack, the more effective key diversity will be in thwarting (or deterring) him. For instance, if an attacker considers worthwhile an attack with 20% chances of success, the effort required with two diverse keys will be more than 3 times greater than with just one key. This increases to almost 10 times greater if the adversary is willing to attack for a 2% probability of success; or for 20% probability of success if

[3] We observe that this is the best strategy if the attacker *has a way of knowing* when he has found the right key, which is the case in the design assumed here: the plaintext contains redundant information, used by the legitimate recipient to verify authenticity. We have considered alternative designs without this property, but their discussion is beyond the scope of this paper.

[4] From the viewpoint of cryptanalytic effort in this attack scenario. From other viewpoints, e.g. physical security against access that allows the adversary to tap signals, multiple cables may or may not increase risk, depending on details of geometry, etc. We expect, however, that the degree of replication of communication channels will be dictated by reliability and safety requirements, and will thus be a given for the designers of the encryption mechanisms.

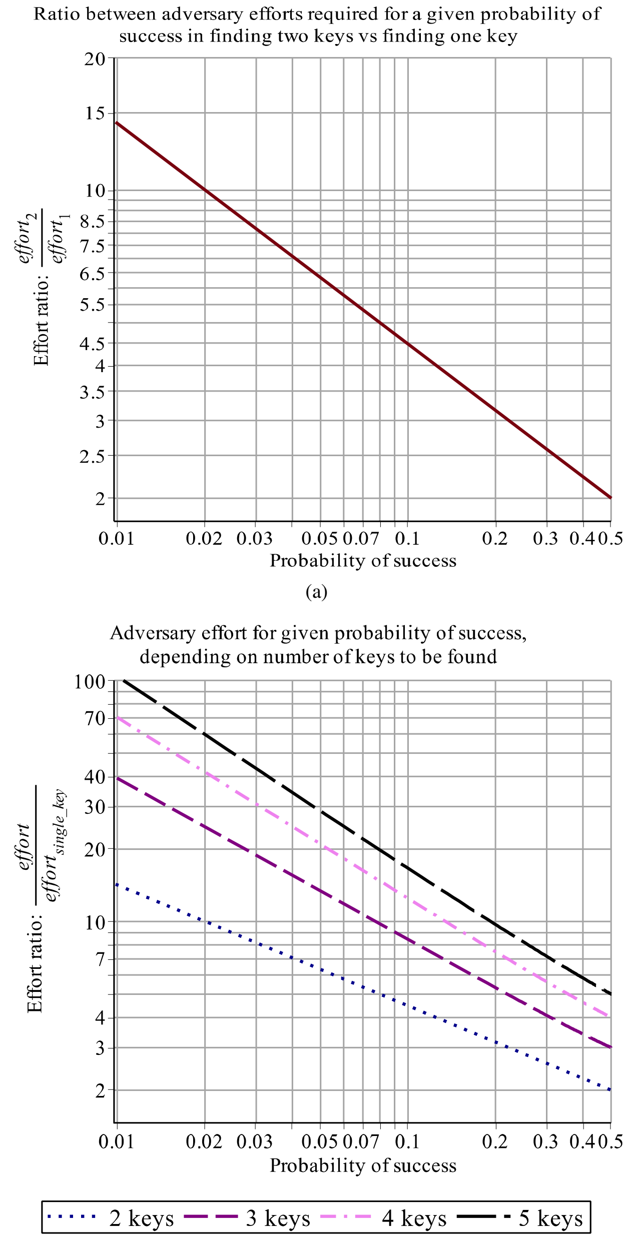
[5] These plots are calculated assuming a small key space of 1000 just for the sake of illustration, to plot results as a function of *t*, requiring an example value of T_{exh}; but the curves are to all practical effect identical for any larger key-space.

the attacker needs to find five keys instead of just one key[6]. In the worst case that the adversary can afford a complete search, of course key diversity still makes *certainty* of success twice as expensive for two keys, or m times as expensive for m keys; likewise, the mean effort required for success on m keys is m times the mean for one key, as the sum of m identically distributed, independent random variables.

Seen from the designers' viewpoint, this also means that if their target is a probability of integrity violation of τ, they only need to ensure that this probability is, for each single channel, $\sqrt{2\tau}$. For instance to assure a probability of integrity violation under 10^{-6}, it is sufficient to ensure a probability of violation per channel of $1.41\ 10^{-3}$.

Conversely, however, if the adversary's probability of success on one channel is high, using diverse keys in the different communication channels only brings small advantages. For probabilities of success between 0.5 and 1, the effort required for two keys is between approximately 1.7 and 2 times that required for one key. In other words, a communication architecture that requires the adversary to find two diverse keys substantially strengthens cryptographic protection that is already reasonably strong with a single key, but adds very little strength if not.

In conclusion, key diversity very effectively reduces the risk of integrity violations from random search through a reduced key space, for flaws in the encryption method implementation that give attacker a low but worthwhile (for the attackers) chance of success in such attacks. The concern is then whether flaws exist, unknown to the system designer, that make this probability of success too high for key diversity alone to be a sufficient defence. Then, diversity between the implementation of security in the redundant channels will be desirable. We discuss this scenario in the next section.



(a)

(b)

Fig. 3. Adversary effort caused by diverse keys: increase in effort required to achieve a desired probability of (a) finding two keys vs finding one key; (b) finding m = 2, 3, 4 or 5 different keys, *vs* finding one key.

[6] An architecture requiring the adversary to find several keys in order to succeed in an integrity attack would not need more than three physical communication channels. Given enough bandwidth, it could use multiple encrypted copies of each message, sharing the same channel. There would be drawbacks to avoid: identical copies would give an adversary who has discovered a key a known plaintext for searching other keys; while in the basic configuration we study, readings of different sensors will normally exhibit small random discrepancies, which make it harder for an adversary to exploit the fact that the readings must be physically consistent. We will not explore this design variation further in this paper.

V. "Supply chain" attacks and diversity

In this scenario, the adversary exploits vulnerabilities in the implementation of encryption, introduced intentionally, or present by accident and disclosed to the adversary and not to the defenders, etc. The flaw may be in the implementation by a specific vendor. To reduce this risk, the designer of our control system may choose to procure more than one, diverse versions of the encryption components, from three different manufacturers, to use in the three channels. Or a flaw may be discovered in the cryptography algorithm itself, and secretly sold to the adversary. Against this risk, the designer may want to use different algorithms in the different channels.

Just as in the use of diversity against accidental faults, the designer would try to obtain for the communication channels in the system "substantially different" versions of the cryptography function to be provided, meaning that, in addition to coming with good evidence of quality, they (i) are as different as possible, and (ii) give as good guarantees of having been developed independently as possible.

This diverse procurement brings costs, e.g. in more complex supply chains and maintenance, creating trade-offs that are well known in applications of diversity for safety. The additional costs are justified if protecting high value targets, or large enough numbers of deployed systems.

The requirements about what aspects should be different between diverse versions relate to the set of causes of common failures that one considers to require diversity as a defence despite other precautions [3], [6]. For instance, buying versions from different vendors would reduce the risk of a common flaw in the development cultures behind the versions, which might cause flaws affecting the versions themselves; in view of the fact that similar algorithms to be implemented, or similar structures of hardware or software design, may make the same kind of errors likely for different developers, one would like evidence that the algorithms and structuring of the programs are different; etc. Other parts of the "implementation" of a complete crypto solution also matter: one would want some assurance that the key generation processes for different versions are not only apparently good but also different, reducing the likelihood of shared flaws; for that matter, one might require that crypto chips meant for different communication channels be delivered to the factory by different couriers using different warehouses, if there was a concern that chips might be "borrowed" along the way by the adversaries to study (for instance, by recording their behaviours with a set of known plaintexts) before they are installed in the system.

Let us assume that n "versions" of the implementation (software and hardware) of an encrypted communication channel are available on the market. That is, our system designers, looking for off-the-shelf component implementing encryption with the key length and other general properties they require, can choose between n solutions that are different in all details required (as discussed in the previous paragraph). Out of these, a (generally unknown) number k contains flaws of the very serious ("fatal", henceforth) type about which we are concerned, such that diversity of keys is not a sufficient protection, which are known to adversaries, or will be at a future time of concern. For brevity, we will talk simply of *versions* which may be *flawed* or *correct*. As the simplest scenario, we assume that each version is "as good as" any of the others from the system designer's viewpoint: same cost, cryptographic strength, power consumption, etc., and same probability of having flaws of any given level of severity: therefore the system designers will choose among them randomly (without replacement), with equal probabilities. For a three-channel system, the system designer selects between one and three versions out of the n versions on the market. The probability of a randomly chosen version being flawed is $\frac{k}{n}$, which we call q.

What are the probabilities of the system built being open to attack ("flawed", for brevity), depending on how many versions the system designer decides to use?

With a single version deployed in all three channels, the probability of having a flawed version (and thus three flawed channels) is $Q_{single}=q$. Three identical channels are as vulnerable (both for confidentiality and integrity) as a single channel.

With two versions (one version for one channel and another for the other two),

- the probability of a flaw in at least 1 out of 3 channels is the probability that at least 1 out of 2 versions is vulnerable:

$$Q_{double,1/3} = \frac{2q - q^2 - \frac{q}{n}}{1 - \frac{1}{n}}$$

  which for the (maybe unlikely) case of large n would be approximately $2q$-q^2;

- the probability of a flaw in at least 2 out of 3 channels is the probability that the version that is chosen for two channels is flawed:

$$Q_{double,2/3}=q;$$

Thus using two versions *increases* the risk to confidentiality *without reducing the risk to integrity.*

With three versions (one per channel),

- the probability of a flaw in at least 1 out of 3 channels is :

$$Q_{triple,1/3} = 1 - \frac{(1-q)\left(1-q-\frac{1}{n}\right)\left(1-q-\frac{2}{n}\right)}{\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)}$$

  which for large n would be approximately 1-(1-q)^3;

- the probability of flaws in at least two channels would be

$$Q_{triple,2/3} = \frac{q\left(q - \frac{1}{n}\right)\left(3 - 2q - \frac{2}{n}\right)}{\left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right)}$$

  or, for large n, approximately $3q^2$-$2q^3$.

  Fig 4 plots $Q_{triple,1/3}$ and $Q_{triple,2/3}$ against q.

For k=1 (only one flawed version among the n on the market), diversity in a two-out-of-three configuration makes a system integrity flaw impossible, while it does not eliminate the risk of a system confidentiality flaw, which has probability $\frac{3}{n}$.

This is highlighted in Fig 4 by the points showing the two probabilities for n=5, $\frac{k}{n}$=0.2.

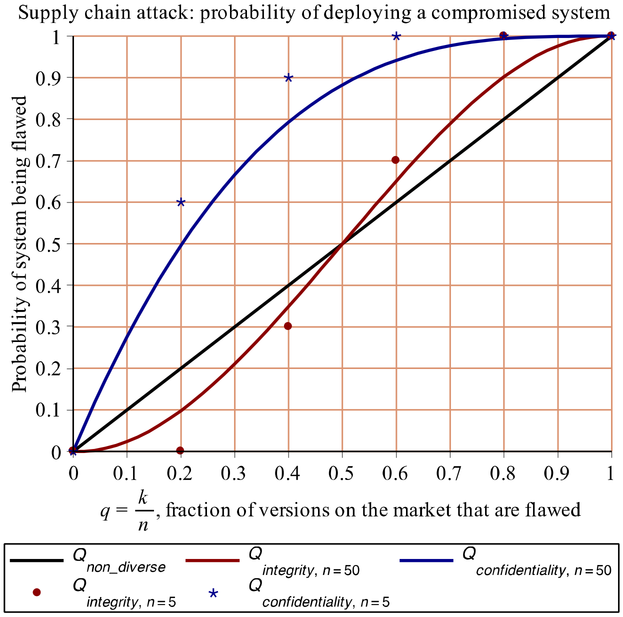


Fig. 4. "Supply-chain" attack: Probabilities of having a compromised system with a non-diverse vs a 3-version implementation, as a function of the probability for a single version. The continuous lines show the shape of the functions for n>>1. The points show an example in which the market offers only few (5) versions (choices of implementation).

Compared with a single version, for q < 0.5 using three diverse versions reduces the probability of integrity breach but increases the probability of confidentiality breach: the well known, common trade-off between confidentiality and integrity. As may be expected by analogy with 2-out-of-3 voted systems, with q>0.5 diversity increases risk[7].

Using diversity seems justified if it reduces the integrity risk by more than it adds to the confidentiality risk. If we define risk as the product of the probability and loss associated with an undesired event, this condition is written as:

$$L_C(Q_{triple,2/3} - q) < L_I(q - Q_{triple,1/3}).$$

where L_I is the expected loss from an "integrity flaw" (building the system with two or more "flawed" channels), L_C is the expected loss from a "confidentiality flaw" (building it with any "flawed" channel). For q close to zero (few flawed versions out of a large number n of available versions), this inequality reduces approximately (and exactly when k=1) to:

$$2L_C < L_I,$$

The threshold may however be quite higher than 2 if a substantial fraction of the "each as good as the others" versions, available for the designers to choose from, are flawed. This is illustrated in Fig 5. Since diversity is never desirable for $k \geq (n/2)$, the horizontal axis is truncated at q=0.5. We can see that for instance (rightmost round marker in the plots) if n=7 and k=3 of the 7 available versions are flawed, diversity is a correct risk reduction choice if $\frac{L_I}{L_C}$ > 8. For large values of $\frac{q}{n}$, the threshold for $\frac{L_I}{L_C}$ is even higher. However, for many embedded systems the ratio $\frac{L_I}{L_C}$ will be above the threshold, as attacks on integrity can directly lead to costly accidents. That is, diversity among the encrypted channels will usually be the preferred option, in those uses of this control system for which sophisticated attackers would be interested in sabotage.

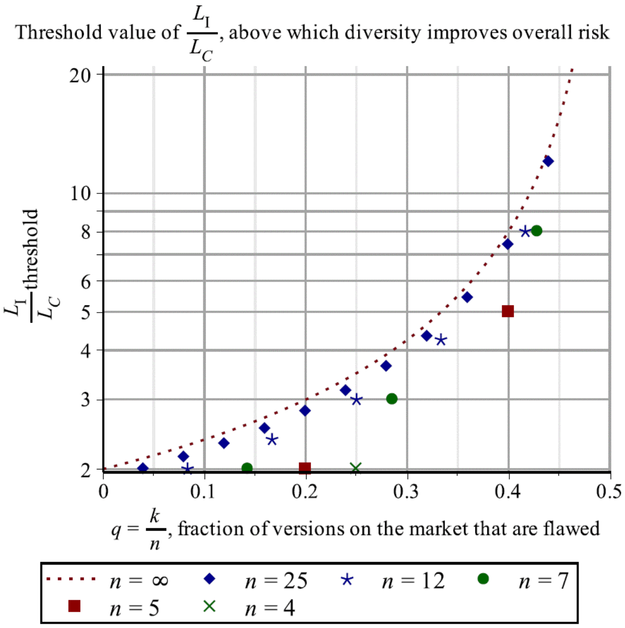


Fig. 5. Conditions under which diversity between crypto implementations in the three channels reduces, on balance, the risk from "supply chain" attacks. For this to be the case, the ratio L_I / L_C needs to be above the curve shown, for the assumed value of q (the probability that the first version randomly chosen from those available has fatal security flaws known to the adversary).

VI. Discussion and Related Work

The area of redundancy and diversity for security has been studied by various authors (cf e.g. an early survey and references [3]; a more recent one in [7]). Apart from the various proposals of architectures using them (usually with limited and often simplistic modelling; see the discussion in [3]) there is empirical work on the effectiveness of diversity and defence in depth with specific products [8], [9], [7], [10]

[7] The curves in Fig 4 qualitatively resemble the well-known curves for a 2-out-of-3 voted system and for a "series" system of 3 components, with the difference that here the k versions are *sampled without replacement* from the n available, so that e.g. channel A being flawed and channel B being flawed are not strictly independent events, for any finite n. So the distribution of the number of flawed versions in the system is hypergeometric.

Measure of adversary effort [4] have become accepted in very different areas [11] and are generally present in studies of cryptography, although it is common to report simple measures like mean times to success, while we study the probability of success as a function of effort, to derive interesting properties.

Differently from most of these studies, we explicitly address the issue of driving decisions through probabilistic analysis, in a case in which this is feasible, on a "what if" basis of assuming a threat and assessing alternative designs in view of that threat. This is akin to the approach often used in assessing cryptography. However, our focus here has not been to analyse a specific cryptographic scheme and the performance of possible attacks on it. Rather, we assumed simple models of probability of success on a single key as a function of effort, and we studied the effects of redundant channels and diversity of keys. We used a linear model of probability of success for attacks on each key as a function of the effort spent; this can be replaced by any other models as required.

While we have studied a specific case of communication in embedded systems, where the design parameters will be typically constrained by standard practice of design for safety and reliability in absence of attacks, we believe this is a useful example of how to study design trade-offs for a more general class of problems.

Outside our specific case study of protecting replicated data paths via cryptography, many authors have considered the use of replication and diversity for security, in various contexts. Applications that have been analysed probabilistically are in redundant computations or data storage; researchers have been interested in adapting voting and quorum schemes and Byzantine fault-tolerant protocols to deal with security concerns. However, these typically assess probability of integrity violation (attackers being able to corrupt the result of a vote, or to avoid detection of corrupt behaviour by compromised copies of a process), and trade-offs with cost and performance overheads, or between reliability of voted results in the presence and in the absence of attacks. A few recent examples are in [12], [13], [14], [15], [16].

On the other hand, in our experience practitioners and researchers routinely acknowledge the existence of integrity-confidentiality trade-offs, but surprisingly we have not found formal mathematical analyses as we propose here. In the much more complex scenarios of replicated distributed processing or storage, examples of studies involving confidentiality and security tend to propose architectural schemes and evaluate them in terms of deterministic properties (e.g. of Byzantine fault tolerant protocols and of secret sharing methods) and study the performance overhead imposed by replication and fault-tolerant protocols, and thus performance-security trade-offs. (e.g. [17], [18]).

Study of the security-reliability-safety trade-offs in the design of embedded systems was a main theme of the SeSaMo project (deliverables and project publications are at http://sesamo-project.eu/documents).

From the viewpoint of the modelling assumptions and general results, one may wish to compare these results with those for redundancy and diversity against accidental faults.

- our modelling exploits specific assumptions of independence between the events modelled. Undue assumptions of independence have been shown (by us among others) to be the Achilles's heel of many probabilistic analyses about redundancy and diversity [19], [20]. Here, these assumptions are based on (1) in the cryptanalysis scenario, the keys being assigned independently; (2) in the "supply chain" scenario, the versions being chosen independently conditionally on previous choices (by sampling without replacement). Other scenarios will require more complex assumptions, which we have not studied yet. For instance,
    - the cryptanalysis scenario may change if the first key found gives useful information that reduces the effort for finding the next key;
    - the "supply chain" concerns apply not only to choosing versions that may be flawed *today*, but to exploring the *probability* of their being flawed, or "becoming" flawed (in the sense of vulnerabilities being discovered and made available to adversaries) a few years from now. The processes of creation and discovery of flaws for diverse versions may well be probabilistically dependent.
- we do model a form of dependence between "failures" of the channels due to the fact that the adversary has a single pool of effort for attacks on any channel: effort spent against a channel A is not used for attacking channel B. By contrast, models of accidental failure assume that exposure to failure affects all the redundant channels in parallel. This explains why, in the special case of independence between failures of redundant channels, the standard expression for the probability of accidental failure in a 2-out-of-3 voted system is quadratic in the probability of failure of a single channel ($Q_{2\text{-out-of-3}} \approx 3(Q_{single})^2$), as ours in section IV, but with a larger multiplicative constant.
- the other important difference is that for accidental failures the hazard rate is usually assumed constant or increasing slowly, for at least part of the lifetime of the system, while in our random search scenario the probability of successful breach of one channel increases linearly with effort.

VII. Conclusions and Further Work

We have demonstrated some forms of probabilistic analysis on the security of communication in an embedded system, in which the time span of attacks and the time horizon for security assessment may range from days to decades.

Novel aspects of this study include applying this style of analysis to:

- assessing adversary effort for less-than-certain success, an important aspect for both attackers' and designers' decisions;
- addressing quantitatively the trade-offs between integrity and confidentiality in using redundancy/diversity for security;
- considering these trade-offs with respect to defence against the effects of unknown vulnerabilities, an extremely important concern which however is seldom analysed systematically.

We discuss further down how this style of analysis can be easily extended to many other systems, and in fact our detailed results already apply to other systems and scenarios that resemble ours from the viewpoint of the design properties we modelled. However, a more important conclusion is that our examples demonstrate how probabilistic analysis for a *specific* system can give useful answers about *that* system, including guidance for design decisions and possibly "surprises", such as, in this case, revealing circumstances in which the expected trade-off between confidentiality and integrity does not apply (diversity will improve integrity without damaging confidentiality).

We list some interesting observations that we derived *for this specific design*, and some ways in which they suggest more general insights:

- for a cryptanalysis attack on this system, seeking to obtain keys by random search, the most efficient strategy for an adversary with a finite stock of available effort is to attack one channel, and only after success moving effort to another channel;
- given this fact, using diverse keys on replicated communication channels improves integrity (in particular, defences against sabotage) without harming confidentiality. This may be a surprising finding for a designer who had not attempted this kind of analysis;
- having multiple keys can substantially reduce the risk of integrity violation, that is, of an adversary being able to forge a majority of the redundant messages so as to "hijack" a controlled system to do harm;
- we note that while the degree of replication of physical media is dictated by reliability/cost trade-offs, it will often be feasible, if necessary, to decide the number of diversely encrypted copies of messages to be sent and voted upon (with perhaps more than one copy on each physical link) at the level desired for integrity protection;
- if the adversary needs to spend the amount of effort *t* in order to ensure a given probability Q of a successful attack on integrity of a single version, then to ensure the same probability of successful attack against three different keys he or she has to spend the amount of effort $\sqrt{2tT_{exh}}$ (where T_{exh} is the effort needed to penetrate the protection of a single channel with certainty, i.e., completing the exhaustive search). The difference can be large indeed, and **especially large** if adversaries are willing to attack even for small probabilities of success, as could be the case for **high-value targets**;
- in the supply chain attack scenario, in which the adversary exploits fatal weaknesses unknown to the designer, the analysis confirms the intuition that diversity buys improved integrity at the price of weakening confidentiality, and gives a quantitative estimate of the overall change in risk level and thus a criterion to support decisions. An interesting observation is that the set of circumstances in which diversity is undesirable is broader than one would expect from the analogy with 2-out-of-3 redundancy against random failures; and yet, in the common situation in which integrity flaws (allowing adversaries to cause accidents) are substantially more dangerous than confidentiality ones, **diversity is desirable over a large range of scenarios**.

We have modelled very basic properties of any redundant system subject to attacks. Thus, for instance, the law linking adversary effort to probability of success is characteristic of any situation in which the attacker explores a search space such that each item in it has the same probability of being the "winning" one, and the defender selected these independently from the search space. So, if the design problem concerns, for instance, the probability of an attacker taking control of two servers by random search of a space of authentication credentials (and can recognise when he found the right credentials, so that a "sequential" style of attack is optimal, as in section V), the same law applies.

Just as our models are not limited to a particular kind of redundant component, so they are not limited to a specific redundant architecture. For any set of multiple channels with our probabilistic assumptions on the search space, the optimal strategy for the attacker is to attack one channel at a time (sequentially). So:

- if the redundant system is a self-checking pair, the probability of breach in one communication channel (what we called probability of confidentiality breach) will also be the probability of an availability breach (because it allows the attacker to make the self-checking pair shut itself down), while breaking two keys will allow an integrity breach (that is, making the pair issue consistent incorrect outputs);
- if the redundant system is a 1-out-of-n safety protection system (any one redundant protection lane has authority to perform safety shut-down on a plant), breach of one or two communication channels will allow attackers to perform a spurious shutdown or to reduce fault tolerance against accidental failures; breach of *n* will allow them to prevent a safety shutdown.

Extensions to any number of redundant channels and of keys are elementary; the reduction in probability of success as a function of effort is of the order of τ^m for *m* keys, as shown in the Appendix. However, the details of architectures using more keys, including their reliability and safety properties, require further study.

We have only analysed a very simple model of attack, but we see no conceptual difficulty in extending this style of analysis to more complex system; e.g., using separate cryptographic means for authentication and for confidentiality. The extension to attack models with different laws governing the probability of success as a function of effort is also conceptually simple, although it may be mathematically complex. For instance, we can consider attacks relying on the predictability of certain parts of the plaintext (e.g. message sequence number), or in which success on one key (on one channel) simplifies the process of attacking another key (on another channel). We plan to work on these aspects.

Another direction for future work is more detailed analysis of our scenarios of systematic vulnerabilities in crypto systems. We have described, in essence, thought experiments in which the availability to the attacker of reduced search opportunities (with given size of the reduced search space), or "fatal flaws", is a parameter of the scenario and thus of the design problem. We believe this is a good approach for gaining insight. But for assessing the likely results of a design decision, and the degree of uncertainty about it, more complex reasoning will be required, in which these parameters are themselves the results of a probabilistic analysis of the processes that produce such vulnerabilities. The difficulty of inferring reasonable distributions of model parameters from the available knowledge is the foremost problem for research in security assessment, just as it is in important areas of reliability and safety assessment.

Acknowledgment

This work was supported in part by the SeSaMo project funded by the Artemis JU, the U.K. Technology Strategy Board (ID 600052), and the UK EPSRC project D3S (Diversity and defence in depth for security: a probabilistic approach). We thank Martin Matschnig, Thomas Hinterstoisser and Bernhard Fischer from Siemens for introducing us to the original case study that inspired this work. We thank Peter Ryan for comments on an earlier version of this paper.

References

[1] L. Strigini. "Fault tolerance against design faults," in Dependable Computing Systems: Paradigms, Performance Issues, and Applications, H. Diab and A. Zomaya, Eds. 2005, .

[2] R. T. Wood, R.Belles, M. S. Cetiner, D. E. Holcomb, K. Korsah, A. S. Loebl, G. T. Mays, M. D. Muhlheim, J. A. Mullens, W. P. P. III, A. L. Qualls, J. T.L. Wilson and M. E. Waterman. Diversity strategies for nuclear power plant instrumentation and control systems. NRC, U.S. Nuclear Regulatory Commission. 2010.

[3] B. Littlewood and L. Strigini. Redundancy and diversity in security. Presented at ESORICS (European Symposium on Research in Computer Security). 2004, .

[4] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid and D. Gollmann. Towards operational measures of computer security. Journal of Computer Security 2(3), pp. 211-229. 1993.

[5] I. Gashi, A. A. Povyakalo, L. Strigini, M. Matschnig, T. Hinterstoisser and B. Fischer, "Diversity for safety and security in embedded systems," in Faste Abstracts of the IEEE International Conference on Dependable Systems and Networks, Atlanta, GA, USA, 2014, .

[6] B. Littlewood and L. Strigini. A discussion of practices for enhancing diversity in software designs. Centre for Software Reliability, City University London. 2000.

[7] M. Garcia, A. Bessani, I. Gashi, N. Neves and R. Obelheiro. Analysis of operating system diversity for intrusion tolerance. Software: Practice and Experience pp. n/a-n/a. 2013. . DOI: 10.1002/spe.2180.

[8] I. Gashi, C. Leita, V. Stankovic and O. Thonnard. An experimental study of diversity with off-the-shelf AntiVirus engines. Presented at NCA'09 - International Symposium on Network Computing and Applications. 2009, .

[9] P. Bishop, R. Bloomfield, I. Gashi and V. Stankovic. Diversity for security: A study with off-the-shelf AntiVirus engines. Presented at Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium On. 2011, . DOI: 10.1109/ISSRE.2011.15.

[10] N. Boggs, S. Du and S. J. Stolfo. "Measuring drive-by download defense in depth," in Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden, September 17-19, 2014. Proceedings, A. Stavrou, H. Bos and G. Portokalidis, Eds. 2014, Available: http://dx.doi.org/10.1007/978-3-319-11379-1_9. DOI: 10.1007/978-3-319-11379-1_9.

[11] G. Schudel and B. Wood. Adversary work factor as a metric for information assurance. Presented at New Security Paradigm Workshop. 2000, .

[12] R. Venkatakrishnan and M. A. Vouk. Using redundancy to detect security anomalies: Towards IoT security attack detectors: The internet of things (ubiquity symposium). Ubiquity 2016(January), pp. 1:1-1:19. 2016. Available: http://doi.acm.org/10.1145/2822881. DOI: 10.1145/2822881.

[13] A. Bendahmane, M. Essaaidi, A. El Moussaoui and A. Younes. The effectiveness of reputation-based voting for collusion tolerance in large-scale grids. IEEE Transactions on Dependable and Secure Computing 12(6), pp. 665-674. 2015. . DOI: 10.1109/TDSC.2014.2369049.

[14] L. Wang, S. Ren, B. Korel, K. A. Kwiat and E. Salerno. Improving system reliability against rational attacks under given resources. IEEE Transactions on Systems, Man, and Cybernetics: Systems 44(4), pp. 446-456. 2014. . DOI: 10.1109/TSMC.2013.2263126.

[15] K. Ravindran, A. Adiththan, M. Rabby and J. Jose. Autonomic management of replica voting based data collection systems in malicious environments. Presented at Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks. 2015, Available: http://doi.acm.org/10.1145/2815317.2815319. DOI: 10.1145/2815317.2815319.

[16] J. A. Garay, R. Gennaro, C. Jutla and T. Rabin. Secure distributed storage and retrieval. Theor. Comput. Sci. 243(1-2), pp. 363-389. 2000. Available: http://dx.doi.org/10.1016/S0304-3975(98)00263-1. DOI: 10.1016/S0304-3975(98)00263-1.

[17] R. Padilha and F. Pedone. Belisarius: BFT storage with confidentiality. Presented at Network Computing and Applications (NCA), 2011 10th IEEE International Symposium On. 2011, . DOI: 10.1109/NCA.2011.15.

[18] M. Castro and B. Liskov. Practical byzantine fault tolerance and proactive recovery. ACM Trans.Comput.Syst. 20(4), pp. 398-461. 2002. Available: http://doi.acm.org/10.1145/571637.571640. DOI: 10.1145/571637.571640.

[19] B. Littlewood, P. Popov and L. Strigini. Modelling software design diversity - a review. ACM CS 33(2), pp. 177-208. 2001.

[20] K. Salako and L. Strigini. When does 'Diversity' in development reduce common failures? insights from probabilistic modelling. IEEE TDSC 11(2), pp. 193-206. 2014. . DOI: http://doi.ieeecomputersociety.org/10.1109/TDSC.2013.32.

Appendix: Proofs for Cryptanalysis Attack

1) *Assumptions and notation*

The unit of measurement of effort is the amount of effort needed to check one key, assumed constant over the keys (if it were variable, but the sequence of keys tried is such that the average over a sequence much shorter than t attempts is approximately constant, this average can be the unit of measurement). The total number of keys to search is T_{exh}, and the adversary has a stock of effort t. When convenient, we will use a normalized measure of effort, $\tau = \frac{t}{T_{exh}}$.

The keys are assumed chosen by the defender independently for the various channels. The adversary searches the reduced key space through a sequence of possible keys that is independent of the actual key.

2) *Optimality of sequential attack*

We call "sequential" attack the case in which the adversary attempts to find the key for one channel and, if he succeeds, he uses the remainder of his available effort for finding another key. The calculated probabilities of success below show that this is superior to the alternative of allocating fixed fractions of effort to attacks on three channels in the hope of finding two keys. An intuitive explanation follows: if after spending r_1 attempts on channel 1, r_2 on channel 2, etc., no key has been found, the probability of finding a key at the next attempt on channel i is

1/(T_i-r_i): highest for the channel on which the most attempts have been spent. This is true even at the start, after just one unsuccessful attempt: to maximize the probability of success, one must keep trying on the same channel until successful.

3) Sequential attack: probability of success

The adversary attempts to find the key for one channel. Within the *t* attempts which he can perform with his stock of effort, he may succeeds or not. If he succeeds, (i) he has succeeded in a "confidentiality breach" (he can steal information from that channel); and (ii) if he succeed at the *r*-th attempt, *r*<*t*, he can use the effort left in his stock for (*t*-*r*) attempts to find the key for a second channel, seeking an "integrity breach" (that is, to be able to highjack the control function via a 2-out-of-3 majority of forged messages).

The probability of any given key being the correct one is 1/T_{exh}, hence the probability $Q_1(t)$ of finding the correct key for one channel in *r* attempts is:

$$Q_1(r) = \begin{cases} \frac{r}{T_{exh}} \text{ if } 0 \leq r \leq T_{exh} \\ 1 \text{ if } r \geq T_{exh} \end{cases}$$

(*r*≥T_{exh} could be a realistic scenario if the adversary only has to search through a reduced search space, as we assume.)

The event "two keys are found within *t* attempts" is the union of mutually exclusive events of the form "one key is found at attempt *r* and the second key is found within the remaining (*t*-*r*) attempts that the adversary can afford". Thus its probability is:

$$Q_2(t) = \sum_{i=1}^{\min(t,T_{exh})} P(1\ key\ found\ at\ attempt\ i)\ Q_1(t-i)$$

When $t < T_{exh}$ this expression becomes:

$$Q_2'(t) = \sum_{i=1}^{t} \frac{1}{T_{exh}} \frac{t-i}{T_{exh}} = \frac{t(t-1)}{2\,T_{exh}^2}$$

Note: if keys are assigned to channels by sampling *without replacement* (no two channels receive the same key), this beco/mes the (usually) practically identical expression:

$$\sum_{i=1}^{t} \frac{1}{T_{exh}} \frac{t-i}{T_{exh}-1} = \frac{t(t-1)}{2\,T_{exh}(T_{exh}-1)}$$

4) Generalization: sequential attack on multiple keys

For the probability of finding *s* keys in *t* attempts, these expressions generalize as:

$$Q_s(t) = \sum_{i=1}^{\min(t,T_{exh})} P(1\ key\ found\ at\ attempt\ i)\ Q_{s-1}(t-i)$$

Which for $t < T_{exh}$ becomes:

$$Q_s'(t) = \sum_{i=1}^{t} \frac{1}{T_{exh}} Q_{s-1}(t-i) = \frac{t!}{s!(t-s)!T_{exh}^s} = \binom{t}{s} {T_{exh}}^{-s}$$

or, for keys assigned without replacement:

$$\binom{t}{s} \frac{(T_{exh}-s)!}{T_{exh}!}$$

5) Parallel attack

The adversary allocates to each channel an amount of effort $t_i \geq 0$ where $i = 1..3, t_i \leq t$ and $t_1 + t_2 + t_3 = t$. For a given choice of these three amounts of effort, the probabilities of the keys being found are, respectively for the three channels:

$$\frac{t_1}{T_{exh}},\ \frac{t_2}{T_{exh}},\ \frac{t - t_1 - t_2}{T_{exh}}.$$

or using normalized effort:

$$\tau_1,\ \ \tau_2,\ \ (\tau - \tau_1 - \tau_2)$$

Since the events "the key is found" for each one of the three channels are independent, the standard analysis for a 2-out-of-3 system with independent failures applies, as follows.

Confidentiality breach. The probability of at least one key being found (probability of confidentiality breach) is:

$$Q_{triple,1/3} = 1 - (1 - \tau_1)(1 - \tau_2)(1 - \tau + \tau_1 + \tau_2)$$

a function that is maximized if two out of the three τ_i are 0 and the third one equals τ.

Integrity breach. The probability of at least 2 keys being discovered is

$$Q_{triple,2/3} = \tau_I\tau_2 + \tau_I\tau_3 + \tau_2\tau_3 - 2\tau_I\tau_2\tau_3$$

or after setting $\tau_3 = \tau - \tau_1 - \tau_2$:

$$Q_{triple,2/3} = \tau_2 + (\tau - \tau_1 - \tau_2)(\tau_1 + \tau_2 - 2\tau_1\tau_2)$$

This function reaches its maximum when

$$\tau_1 = \tau_2 = \tau/3,$$

i.e. when the adversary allocates his/her effort τ uniformly between the three channels, and this maximum value is: $\frac{\tau^2}{3} - \frac{2\tau^3}{27}$

or, for small τ, approximately $\tau^2/3$. Fig 6 shows a contour plot of $Q_{triple,2/3}$.

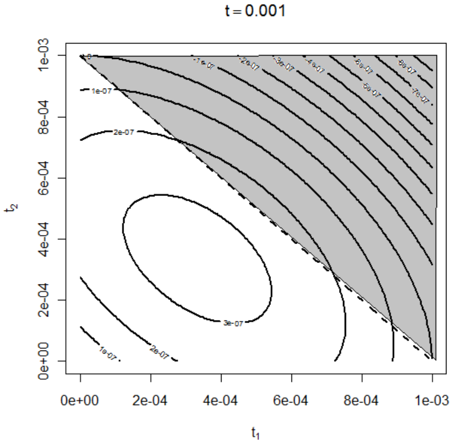


Fig 6. Contour plot of the function $Q_{triple,2/3}$ for $\tau = 0.001$. The points in the greyed out area are not feasible, due to the constraint: $\tau_1 + \tau_2 \leq \tau$.