Gashi, I., Povyakalo, A. A., Strigini, L., Matschnig, M, Hinterstoisser, T & Fischer, B (2014). Diversity for Safety and Security in Embedded Systems. Paper presented at the IEEE International Conference on Dependable Systems and Networks, 23-06-2014 - 26-06-2014, Atlanta, GA, USA.

Diversity for Safety and Security in Embedded Systems

Ilir Gashi, Andrey Povyakalo, Lorenzo Strigini
Centre for Software Reliability
City University London
London, United Kingdom
{i.gashi, l.strigini, andrey}@csr.city.ac.uk

Martin Matschnig, Thomas Hinterstoisser
Bernhard Fischer
Siemens AG, Vienna, Austria
{martin.matschnig, thomas.hinterstoisser,
bernhard.bf.fischer}@siemens.com

***Abstract*— We present ongoing work about how security and safety properties in embedded systems are affected by redundancy and diversity. The need to consider security requirements in the presence of malicious action creates additional design trade-offs besides those familiar in the design of safety critical and highly reliable systems. We outline the motivation for this work, an industrial case study, and the research direction we have taken.**

Keywords - security assessment; safety assessment, safety vs security trade-offs; embdedded systems; software and hardware diversity.

I. BACKGROUND

We address the integration of *security* and *safety* concerns for embedded systems built with *diverse, redundant* components. In critical embedded systems, it is common to apply, for reliability and for safety [1, 2]:

- *redundancy*: using more than one functional modules (performing the same function, from a system-level viewpoint, though possibly in different ways, or some of them monitoring others), to improve the likelihood that the function is performed correctly, or avoids unsafe failures, even if one of them fails
- *diversity*: intentional differences between redundant components, to reduce the likelihood of common failures due to systematic causes (digital design defects common to identical replicas, low reliability under certain environmental conditions, etc.) that would reduce the benefit of redundancy.

Redundancy and diversity also have applications for security, e.g. via redundant defenses or redundant assets; a designer has to anticipate their effects with respect to all faults, accidental or malicious, and any design trade-offs arising. A challenge is in that considering a greater variety of threats or faults may make any analysis very complex. In analyzing redundancy alone ("is there an appropriate element of redundancy against each failure condition for which one is required?") – typically the first step in design analysis – one also needs to add to his model of the system additional interfaces through which adversaries may operate. Then, looking at probabilities of common failures among redundant components, one has to deal with separate parameters for the probabilities of common failure with respect to different causes – accidental or malicious or combinations of both – with potentially very different values.

A. *System Level* Safety *Requirements and Role of Security*

We take as a simple example a 1-out-of-2 diverse system, e.g. a controller for some electro-mechanical system: if one out of two diverse "channels" fails, the other channel is able to detect the failure and trigger a transition to a failsafe state.

Channel failures can be caused by:

1. Accidental hardware faults;
2. Accidental design (hardware, software) faults in the two channels , activated by normal operational/maintenance signals/actions via normal interface(s)
3. Software faults introduced by an adversary
   - During the design / implementation
   - During maintenance (e.g. by installing rootkits), through physical access to the channel
   - Via network interfaces for configuration / maintenance
4. Hardware design faults introduced by an adversary via physical access to the unit (e.g. by re-wiring hardware)
5. Attacks, exploiting accidental design faults, via network interfaces meant for normal operation or maintenance
6. Combinations of cases 1 through 5

For a safety requirement "no accident shall be caused by this controller", the chances of it being satisfied during operation depend on the probability of both channels failing together, due to any combination of these causes.

The security viewpoint here is about *security for safety*: an adversary may produce an accident, or make it more likely, by means 3-6 above. In security terminology, this creates *integrity* requirements (we want the adversary not to be able to cause these failures). Violations of integrity matter because they may cause accidents: an adversary may succeed in making both channels behave unsafely; or at least making one of the channels unable to react properly when the other one fails accidentally (a delayed-effect, "stealthy" attack), making the system effectively non-redundant and thus less safe than is required.

So far, (i) introducing security consideration has not changed the characterisation of this system as a 1-out-of-N system (the system only fails if all channels fail, for any reason); but (ii) the probabilities of common critical failure may vary between the three possible combinations: both channels fail for accidental reasons, both due to malicious action, or one by accident and one through malice. Estimating these probabilities to compare design options, notoriously hard even with accidental faults, becomes harder.

B. *System-level* Security *Requirements*

"Security for safety", the need to prevent adversaries from endangering the reliability of safety functions, is often central in embedded systems. But a system may also have "security-only" requirements. For example, what if for the above system there are *confidentiality* requirements? That is, the two channels contain some information asset A, and A becoming known to an adversary would have no direct effect on data/system integrity, but would be a loss. Confidentiality requirements may arise:

- independently of the safety requirements for the individual system compromised, e.g. being aimed at safeguarding intellectual property in a channel's software;
- or from safety concerns, *indirectly*: e.g., by reading I/O data or the code, the adversary might devise better attacks;
- or, even more indirectly, from a concern that by learning about details of a channel in a certain application context, the adversary gains information for attacks in different applications of the same component.

Details are important. E.g., if A can be obtained through either channel, then our example system behaves – from the confidentiality viewpoint – as a "series" or "2 out of 2" system: compromising one channel compromises the whole. Adding more redundant channels would decrease risk due to accidental faults, *but* typically *increase* risk of violation of confidentiality. *Trade-offs* arise between confidentiality and safety and between direct and indirect safety risk; there is a number of channels that minimises total 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.

Many such complex scenarios are possible even for simple systems. This is the motivation for a case study in the SeSaMo (Security and Safety Modelling for Embedded systems) project, which studies synergies and trade-offs between security and safety through concrete examples.

II. Case Study

Siemens AG has provided to the SeSaMo project a use case for an Industrial Drive application, specifically, motion control in industrial automation and control. Motion control products cover a large variety of variable frequency inverters for synchronous and asynchronous motors, ranging from standard electric motor systems and servomotors (including linear and torque motors) to motors for use in hazardous explosion areas, and customized electric motor systems.

An example of replication within the Industrial Drive application concerns sensor data transmission: three replicated rotary sensors on the motor axis send their readings for processing and adjudication by an algorithm running on an embedded processor within a Field Programmable Gate Array (FPGA) board (Figure 1).

We are studying the design parameters for this case study in view of the synergies and trade-offs between safety and security requirements. Some (inter-related) parameters of interest, and questions we study, are:

- *degree of redundancy* (this is 3 in Fig. 1, but we consider the effects of varying it on safety and security) and *kinds of diversity* (algorithm, implementation, data etc.);
- *adjudication mechanisms*: do the various voting and adjudication algorithms known (e.g. simple majority voting, median voting, etc.) differ in their security characteristics? Could the same attack result in different failure modes and losses depending on the adjudication algorithm? Should the latter be "tunable" to match the attack assumptions in each specific environment where the drive is deployed?
- *the attack modes possible*: aspects of system design will make some attacks likely or unlikely or incredible (e.g. armoured shielded cable will prevent some Electro-Magnetic Interference attacks) and thus contribute to determine the safety-security effects of the redundant items on the system.

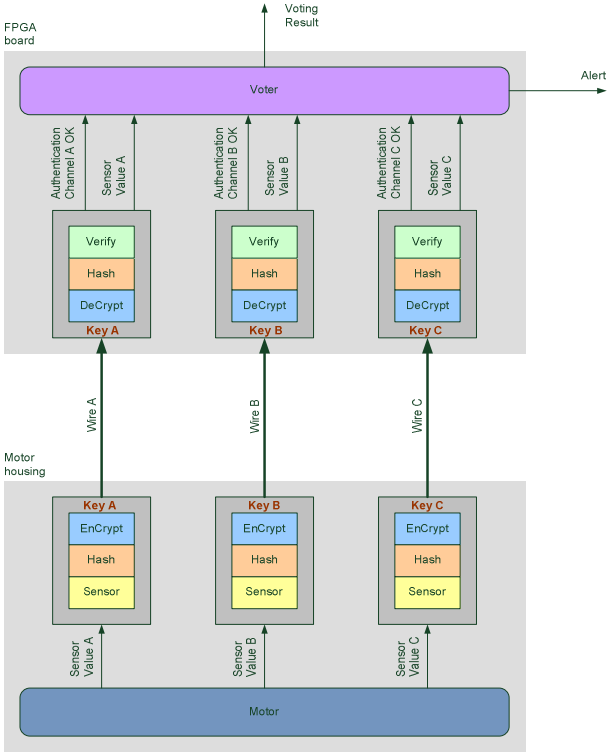

Figure 1 - Replication within the Industrial Drives

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 Austrian grant 834132.

References

[1] L. Strigini, "Fault Tolerance Against Design Faults," in *Dependable Computing Systems: Paradigms, Performance Issues, and Applications,*, H. Diab and A. Zomaya, Eds., ed: J. Wiley & Sons, 2005, pp. 213-241.

[2] R. T. Wood, R.Belles, M. S. Cetiner, D. E. Holcomb, K. Korsah, A. S. Loebl, *et al.*, "Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems," NRC, U.S. Nuclear Regulatory Commission, NUREG/CR 7007, 2010.