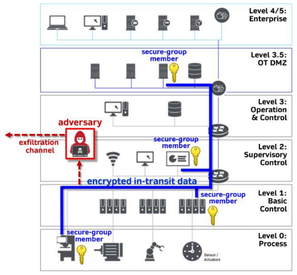
# Risk-Based Optimization of Encryption Cryptoperiods for More Secure ICSs

Gabriele Cianfarani  Natalija Vlajic

Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

## DATA SIPHONING

**Industrial Control Systems** (ICS) play a vital role in managing industrial processes efficiently and safely. Environments contain a **multitude of components** which need to communicate with one another. This increase in interconnectivity means that **secure** and **efficient** communication is critical.

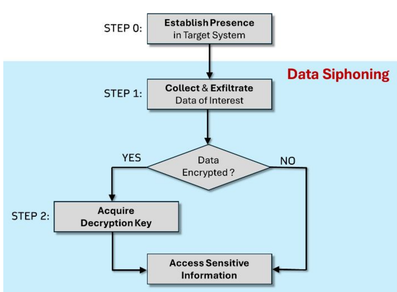



### What is Data Exfiltration?

**Internal network** and **system reconnaissance** is one of the first crucial stages of most cyber attacks

- Involves the **unauthorized transfer** *(exfiltration)* of important operational **in-transit data**
- Many modern ICSs use **encryption**, preventing immediate adversary access to exfiltrated data

Exfiltration must be accompanied with the **acquisition** of the respective ***encryption key***.




### What is Data Siphoning?

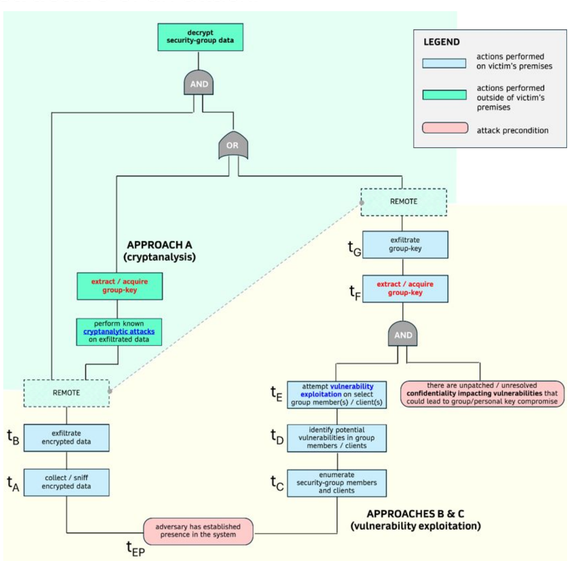
**Data Siphoning** refers to the **entire sequence of steps** that an adversary needs to perform to disclose certain **sensitive information** of interest.

**STEP 1) collection** and **exfiltration** of the given in-transit data

**STEP 2)** efforts to acquire the respective **decryption key** if/when the data from *STEP 1* happens to be encrypted

## ATTACK TREE

An attack tree is a **bottom-up** diagram providing a bird's-eye view of the **logic** and **structure** of an attack.




### What are the main methods of attack?

- **Approach A:** Perform **cryptanalysis** on the siphoned data
- **Approach B:** Gain control of a device through **exploitation** of a:
  - **software vulnerability**
  - **procedural vulnerability**
- **Approach C:** Obtain the key from the Security Key Service *(SKS)* by **impersonating** a legitimate client

**Approach A** generally takes place **outside** the victim's system, while **B and C** must be done **on-site**.

To minimize the impact of Data Siphoning attacks, various **risk-** and **loss-** minimizing measures can be deployed

- **NIST SP 800-82r3** and **NIST SP 800-57** standards suggest periodic key rotation - limiting the respective **cryptoperiod**

## CRYPTOPERIODS

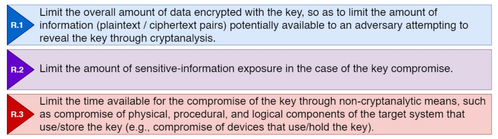
**NIST SP 800.57** | A cryptoperiod is the time span during which a specific key is authorized for use.

### How do you calculate the cryptoperiod?

There are **no** explicit guidelines on how to determine the cryptoperiod in a real-world system.

Some existing literature tackles the problem in **general terms**:

- Cryptoperiods should range from **days to weeks for large data** volumes and **up to two years for smaller volumes**
- The *actual determination* of cryptoperiods should involve the **risk** and **consequences** of *key exposure*




While no *tangible* calculation is given, one thing is clear. **Risk** is the underlying factor which affects a cryptoperiod.

- In **higher risk** environments, the cryptoperiod should become **shorter** and **vice versa**

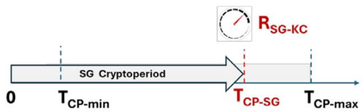
We have developed a **framework** and a respective **software tool** for **A**utomated **R**isk-Based **C**ryptoperiod **C**alculation.

## AUTOMATED RISK-BASED CRYPTOPERIOD CALCULATION (ARC-C)

The **framework** and **software tool** are designed to determine the **optimal cryptoperiod** for each scenario, ensuring a balance between the specific **security risks** and the **operational constraints** of the system.

## FRAMEWORK

The first step is to establish the **upper** and **lower** *'operational feasibility'* limit on the cryptoperiod.




The **cryptoperiod** *($T_{CP-SG}$)* is determined by the estimated risk of the environment, following an **exponential relationship**.

The risk *($R_{SG-KC}$)* can be calculated using: $R_{SG-KC} = P_{SG-KC-succ} \times AI_{SG-KC}$

- $P_{SG-KC-succ}$ is the **probability** that an adversary compromises a key
- $AI_{SG-KC}$ is the **adverse impact** incurred on an organisation if an attack were to succeed

### $P_{SG-KC-succ}$ calculation:

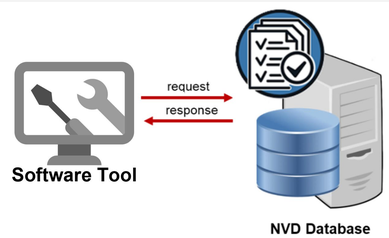
Our **attack tree** model implies two possible approaches to compromise an encryption key:

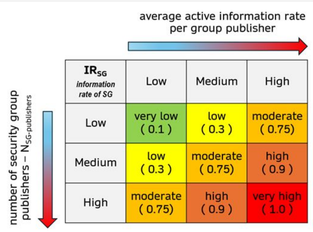
- via **software** vulnerabilities
- via **procedural** vulnerabilities

### $AI_{SG-KC}$ calculation:

The **adverse impact** consists of:

- **Information Rate** or *"how much data is being sent"*
- **Functional Importance** or *"how important is the data"*



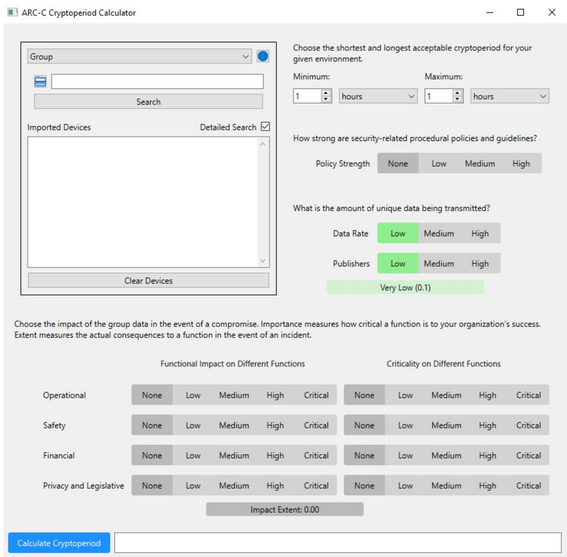

| $IR_{SG}$ information rate of SG | Low | Medium | High |
|---|---|---|---|
| Low | very low ( 0.1 ) | low ( 0.3 ) | moderate ( 0.75) |
| Medium | low ( 0.3 ) | moderate ( 0.75) | high ( 0.9 ) |
| High | moderate ( 0.75) | high ( 0.9 ) | very high ( 1.0 ) |

- The **software probability** utilises the **NVD API** for its calculation
- The remaining variables are **set by the operator**, who selects **qualitative values** via our user interface

## SOFTWARE TOOL

A **software tool** has been developed, combining our framework with a **user-friendly interface.**

- Parameters are grouped to ensure **logical flow**, with each aspect of calculation **self-contained**




To test the **validity** of the framework, we created two hypothetical but highly plausible ICSs:

- **Water Treatment Facility**
- **Energy Storage System** (ESS)

For a more **in-depth** look and a deep dive into the **experimental results**, *scan the QR code.*