

Undetectable Manipulation of CRC Checksums for Communication and Data Storage

Frank Schiller, Tina Mattes
Institute of Information Technology in Mechanical Engineering
Munich University of Technology, Germany

Uwe Weber, Rainer Mattes
Siemens AG, Industry Sector, Nuremberg, Germany

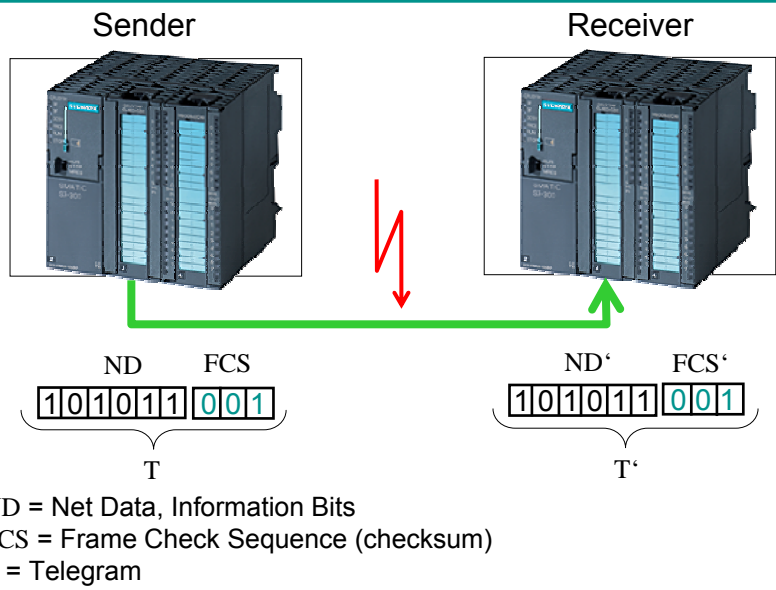
ChinacomBiz'08, August, 28, 2008, Hangzhou, P.R. China

Undetectable Manipulation of CRC

- **Introduction: Application of CRC in Industry**
 - for Communication
 - for Data Storage
- **Some Fundamentals of CRC**
- **Improper Applications of CRC – because of possible**
 - Manipulation for Consistency
 - Manipulation for Consistent Identical Checksum
- **Conclusions and Future Work**

CRC = Cyclic Redundancy Check / Code

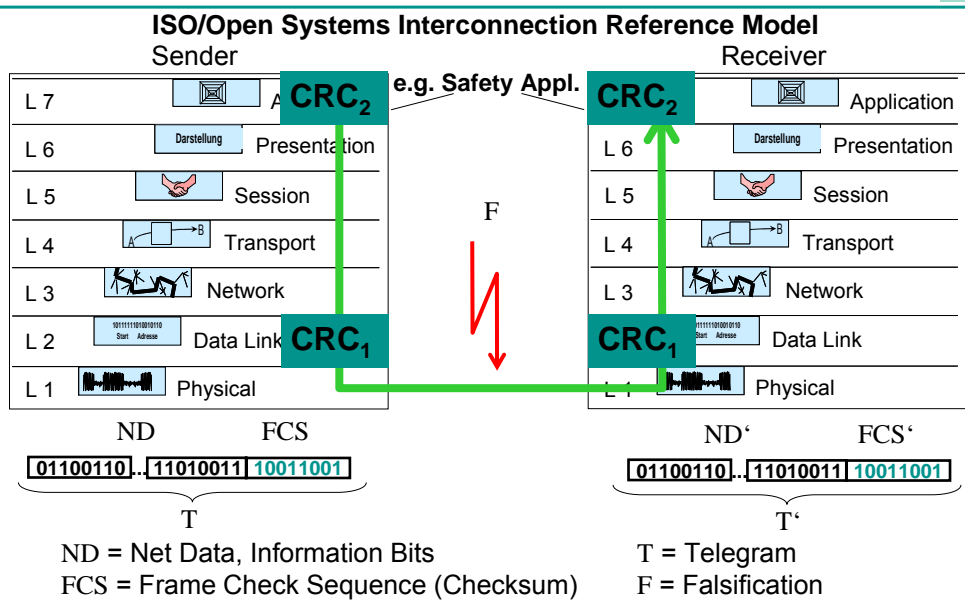
Application of CRC for Industrial Communication



Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #3

Application of CRC for Industrial Communication

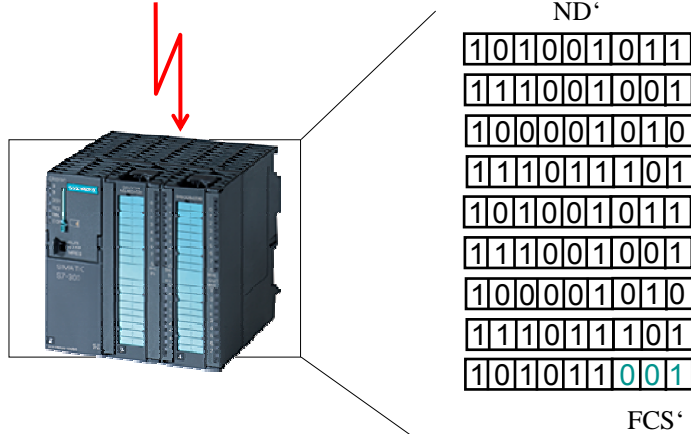


Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #4

Application of CRC for Data Storage

- data or software memory (ROM) within a device



ND = Net Data, Information Bits
FCS = Frame Check Sequence (checksum)

Application of CRC

- for Communication:
 - detection of *random errors* by inconsistency between ND' and FCS'
 - probabilistic criteria: residual error probability
 - deterministic criteria: Hamming-Distance, detectability of burst errors, of inverted telegrams, of specific bit patterns, ...
 - **not** applied for detection of *intelligent errors* since consistency can be achieved again

Application of CRC



- for Communication:
 - detection of *random errors* by inconsistency between ND' and FCS'
 - probabilistic criteria: residual error probability
 - deterministic criteria: Hamming-Distance, detectability of burst errors, of inverted telegrams, of specific bit patterns, ...
 - **not** applied for detection of *intelligent errors* since consistency can be achieved again
- for Data Storage:
 - detection of *random errors* by inconsistency between ND' and FCS'
 - detection of the use of an incorrect version of the software or the data sets by comparing the FCS' with FCS in the documentation!

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #7

Application of CRC



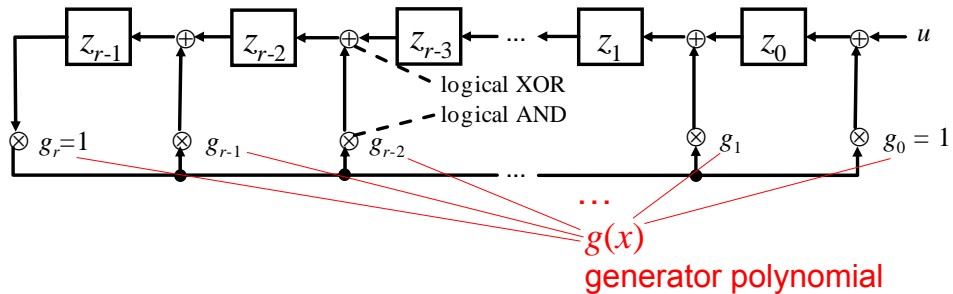
- for Communication:
 - detection of *random errors* by inconsistency between ND' and FCS'
 - probabilistic criteria: residual error probability
 - deterministic criteria: Hamming-Distance, detectability of burst errors, of inverted telegrams, of specific bit patterns, ...
 - **not** applied for detection of *intelligent errors* since consistency can be achieved again
- for Data Storage:
 - detection of *random errors* by inconsistency between ND' and FCS'
 - detection of the use of an incorrect version of the software or the data sets by comparing the FCS' with FCS in the documentation!
 - detection of manipulations of software or data since it would cause a different $FCS' \neq FCS!$ – but a consistent $FCS' = FCS$ can be achieved again

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #8

Some Fundamentals of CRC

- Realization by means of a Linear Feedback Shift Register



- or a sophisticated and efficient table method (cf. literature)

Some Fundamentals of CRC

- In CRC, bit patterns are interpreted as binary polynomials.

Sender

$$(nd(x) \cdot x^r) \bmod g(x) = fcs(x)$$

ND **1010111** $\rightarrow nd(x) = 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0$
 $= x^5 + x^3 + x + 1$

$$g(x) = x^3 + x + 1$$

$$(x^5 + x^3 + x + 1) \cdot x^3 \bmod (x^3 + x + 1) = 1 \rightarrow \mathbf{001} \text{ FCS}$$

Telegram

ND	FCS
1010111	001

Receiver

$$(nd'(x) \cdot x^r + fcs'(x)) \bmod g(x) = 0?$$

$$((x^5 + x^3 + x + 1) \cdot x^3 + 1) \bmod (x^3 + x + 1) = 0?$$

Some Fundamentals of CRC

- Modeling of errors by superimposition:

Receiver

T:	1	0	1	0	1	1
T':	1	0	1	1	1	1
F:	0	0	0	1	0	0

$$(nd'(x) \cdot x^r + fcs'(x)) \bmod g(x) = 0?$$

$$(nd(x) \cdot x^r + fcs(x) + f(x)) \bmod g(x) = 0?$$

$$\underbrace{(nd(x) \cdot x^r + fcs(x)) \bmod g(x)}_{=0} + f(x) \bmod g(x) = 0?$$

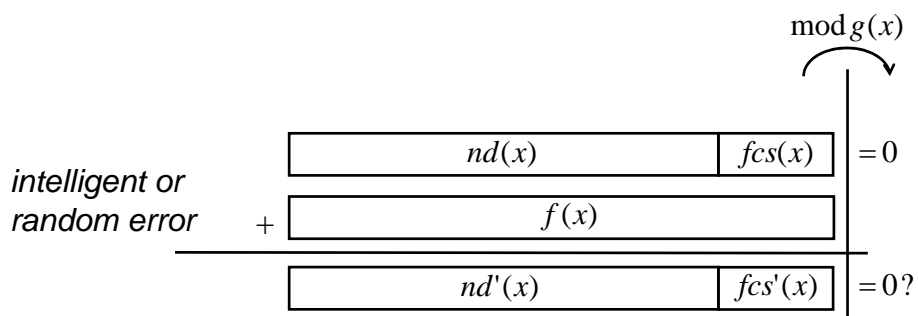


$$f(x) \bmod g(x) = 0?$$

- The analysis is reduced to the check of divisibility of error patterns by the generator polynomial.
- This superimposition can be applied for undetectability of manipulations.

Some Fundamentals of CRC

- Schema of modeling of errors by superimposition:



Undetectable Manipulation of CRC

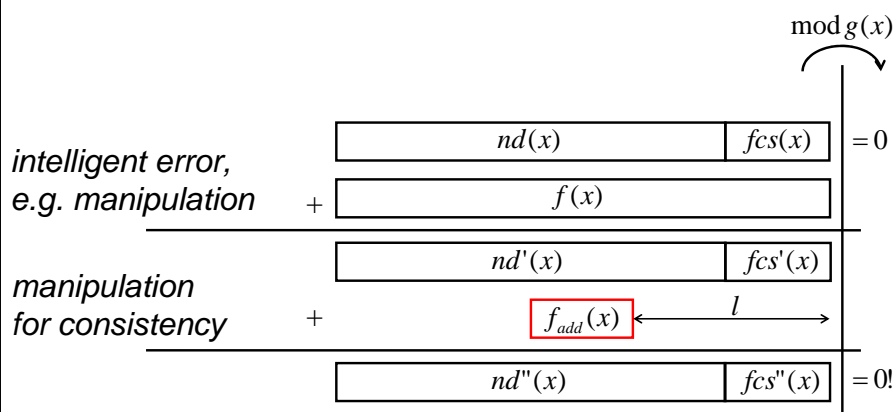


- Introduction: Application of CRC in Industry
 - for Communication
 - for Data Storage
- Some Fundamentals of CRC
- Improper Applications of CRC – because of possible
 - ➔ Manipulation for Consistency
 - Manipulation for Consistent Identical Checksum
- Conclusions and Future Work

Manipulation for Consistency



- Schema of modeling of two errors by superimposition:



Manipulation for Consistency

$$\begin{array}{r}
 \text{intelligent error,} \\
 \text{e.g. manipulation} \\
 \hline
 \begin{array}{r}
 \boxed{nd(x)} \quad \boxed{fcs(x)} \\
 + \quad \boxed{f(x)} \\
 \hline
 \end{array} \\
 \text{manipulation} \\
 \text{for consistency} \\
 \hline
 \begin{array}{r}
 \boxed{nd'(x)} \quad \boxed{fcs'(x)} \\
 + \quad \boxed{f_{add}(x)} \quad \leftarrow l \\
 \hline
 \end{array} \\
 \text{---} \\
 \begin{array}{r}
 \boxed{nd''(x)} \quad \boxed{fcs''(x)} \\
 \hline
 \end{array}
 \end{array}
 \begin{array}{l}
 \text{mod } g(x) \\
 \downarrow \\
 = 0 \\
 \\
 \\
 = 0!
 \end{array}$$

Algorithm 1:

Given: data after main manipulation $nd'(x) \cdot x^r + fcs'$
generator polynomial $g(x)$
shift parameter l

Find: additional error pattern $f_{add}(x)$

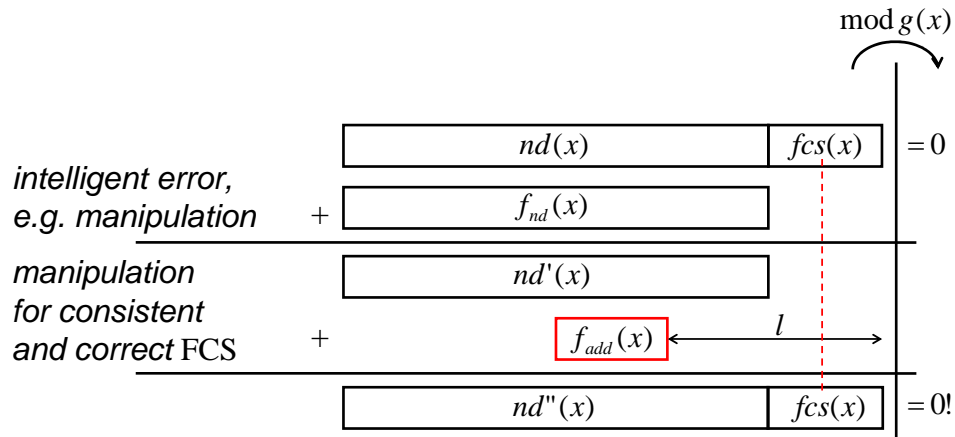
Solution: (see paper)

Undetectable Manipulation of CRC

- **Application of CRC in Industry and Business**
 - for Communication
 - for Data Storage
- **Some Fundamentals of CRC**
- **Improper Applications of CRC – because of possible**
 - Manipulation for Consistency
 - ➔ Manipulation for Consistent Identical Checksum
- **Conclusions and Future Work**

Manipulation for Consistent Identical Checksum

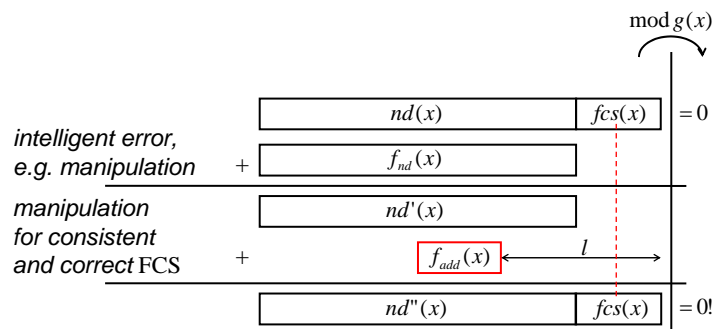
- Schema of modeling of two errors by superimposition:



Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #17

Manipulation for Consistent Identical Checksum



Algorithm 2:

Given: net data after first manipulation $nd'(x)$
 original checksum $fcs(x)$
 generator polynomial $g(x)$
 shift parameter l

Find: additional error pattern $f_{add}(x)$

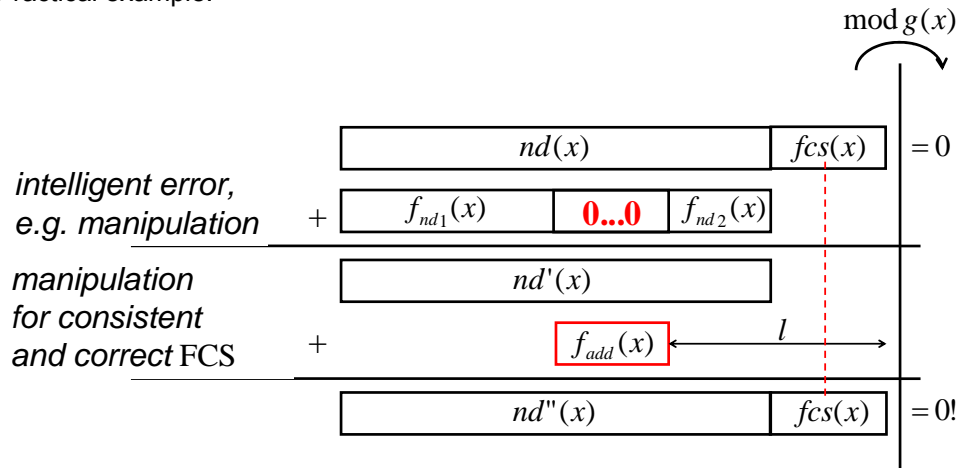
Solution: (see paper)

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #18

Manipulation for Consistent Identical Checksum

- Practical example:



Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #19

Manipulation for Consistent Identical Checksum

- Practical example:

original program	fictitious machine hex code representation
LD x	C442078
LD y	4C442079
-D	2B44
T z	54207A
LD 0xFFFFFFFF	4C44203078FFFFFFFF
LD x	4C442078
PUSH	50555348
+D	2B44
T y	542079
	FCS: 471CE694

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #20

Manipulation for Consistent Identical Checksum

- Practical example:

*intelligent error,
e.g. manipulation*

original program	fictitious machine hex code representation
LD x	C442078
LD y	4C442079
+D	2C44
T z	54207A
LD 0xFFFFFFFF	4C44203078FFFFFFFF
LD x	4C442078
PUSH	50555348
+D	2B44
T y	542079
	FCS: 471CE694

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #21

Manipulation for Consistent Identical Checksum

- Practical example:

*intelligent error,
e.g. manipulation
manipulation
for consistent
and correct FCS*

original program	fictitious machine hex code representation
LD x	C442078
LD y	4C442079
+D	2C44
T z	54207A
LD 0x417FB813	4C44203078417FB813
LD x	4C442078
PUSH	50555348
+D	2B44
T y	542079
	FCS: 471CE694

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #22

Conclusion and Future Work



Conclusion:

- Several non-complex algorithms have been developed for undetectable manipulation of CRC-checksums
- for Communication:
 - **not** applied for detection of *intelligent errors* since consistency can be achieved again
- for Data Storage:
 - detection of the use of an incorrect version of the software or the data sets by comparing the FCS' with FCS in the documentation!
 - detection of manipulations of software or data since it would cause a different FCS' \neq FCS! – but a consistent FCS' = FCS can be achieved again
- Consultancy on the field of reliable, safe, and secure communication is necessary (B2B)

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #23

Conclusion and Future Work



Future work:

- research on CRC based on stochastic automata and dual code
 - find proper generator polynomials
 - even for telegrams/data sets of MBytes
 - analysis of nested CRC for e.g. the use of properties of underlying fieldbus CRC in the safety proof for safety-critical communication
- efficient combination of means for safety and security in industrial communication
- consultancy of small and medium size companies
- collaboration with ITEI, Beijing

Schiller, Mattes, Weber, Mattes: Undetectable Manipulation of CRC

ChinacomBiz'08 #24