Hash Gone Bad:

Automated discovery of protocol attacks that exploit hash function weaknesses






Vincent Cheval[‡] Cas Cremers[*] **Alexander Dax**[*] Charlie Jacomme[‡] Lucca Hirschi[†] Steve Kremer[†]

[*] *CISPA Helmholtz Center for Information Security*

[‡] *INRIA Paris*

[†] *LORIA, INRIA Nancy Grand-Est, Université de Lorraine*

Hashes in security protocols

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

- Communication protocols
  - e.g., TLS, SSH, …

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

- Communication protocols
  - e.g., TLS, SSH, …
- Messengers
  - e.g., Signal, Telegram, …

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

- Communication protocols
  - e.g., TLS, SSH, …
- Messengers
  - e.g., Signal, Telegram, …
- Cryptocurrencies
  - e.g., Bitcoin, Ethereum, …

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

- Communication protocols
  - e.g., TLS, SSH, …
- Messengers
  - e.g., Signal, Telegram, …
- Cryptocurrencies
  - e.g., Bitcoin, Ethereum, …

In security analysis hashes are often assumed to be “perfect”

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

- Communication protocols
  - e.g., TLS, SSH, …
- Messengers
  - e.g., Signal, Telegram, …
- Cryptocurrencies
  - e.g., Bitcoin, Ethereum, …

In security analysis hashes are often assumed to be “perfect”

Random Oracle Model (ROM)

Hashes in security protocols

Cryptographic Hash functions are a basic building block for modern Security protocols

- Communication protocols
  - e.g., TLS, SSH, …
- Messengers
  - e.g., Signal, Telegram, …
- Cryptocurrencies
  - e.g., Bitcoin, Ethereum, …

In security analysis hashes are often assumed to be “perfect”

- meets all desired cryptographic properties
- both in the computational and symbolic setting

Random Oracle Model (ROM)

And in the real world?

And in the real world?

Collisions do exist!

And in the real world?

Collisions do exist!

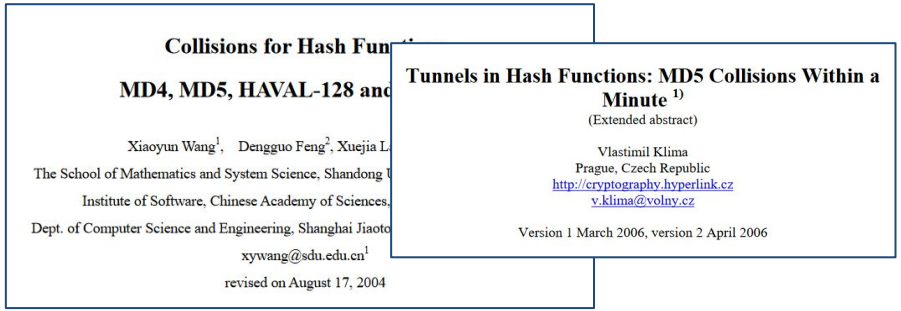
Collisions for Hash Functions

MD4, MD5, HAVAL-128 and RIPEMD

Xiaoyun Wang[1], Dengguo Feng[2], Xuejia Lai[3], Hongbo Yu[1]
The School of Mathematics and System Science, Shandong University, Jinan250100, China[1]
Institute of Software, Chinese Academy of Sciences, Beijing100080, China[2]
Dept. of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China[3]
xywang@sdu.edu.cn[1]
revised on August 17, 2004

And in the real world?

Collisions do exist!



Collisions for Hash Fun

MD4, MD5, HAVAL-128 and

Xiaoyun Wang[1], Dengguo Feng[2], Xuejia L

The School of Mathematics and System Science, Shandong U

Institute of Software, Chinese Academy of Sciences,

Dept. of Computer Science and Engineering, Shanghai Jiaoto

xywang@sdu.edu.cn[1]

revised on August 17, 2004

Tunnels in Hash Functions: MD5 Collisions Within a Minute [1)]

(Extended abstract)

Vlastimil Klima

Prague, Czech Republic

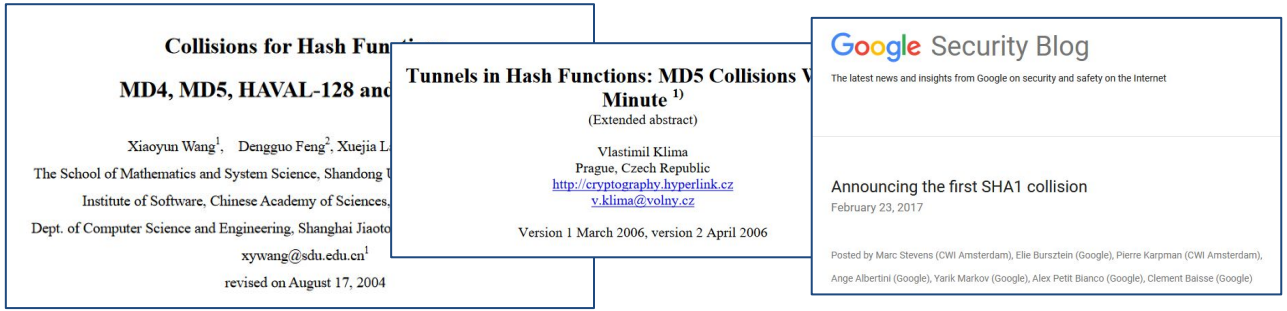
http://cryptography.hyperlink.cz

v.klima@volny.cz

Version 1 March 2006, version 2 April 2006

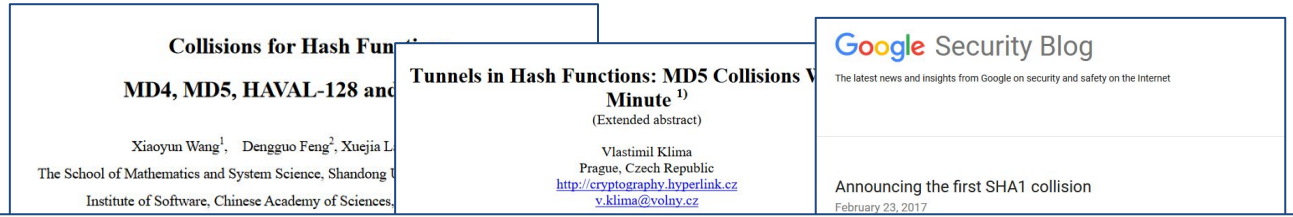
And in the real world?

Collisions do exist!



And in the real world?

Collisions do exist!



Lifetimes of popular cryptographic hashes (the rainbow chart)

Function	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Snefru																												
MD2 (128-bit)[1]																												
MD4																												
MD5															[2]													
RIPEMD															[2]													
HAVAL-128[1]															[2]													
SHA-0																												
SHA-1																												[3]
RIPEMD-160																												
SHA-2 family																		[4]										
SHA-3 (Keccak)																												

Key | Didn't exist/not public | Under peer review | Considered strong | Minor weakness | Weakened | Broken | Collision found

And in the real world?

And in the real world?

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Length Extension (e.g., SHA-1, SHA-2)

- Not a traditional property for cryptographic hashes
- From H(x) an adversary can produce H(x||y) without knowing x
- Example: Breaking authentication in Flickr

flickr

And in the real world?

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Length Extension (e.g., SHA-1, SHA-2)

- Not a traditional property for cryptographic hashes
- From H(x) an adversary can produce H(x||y) without knowing x
- Example: Breaking authentication in Flickr

Transcript Collisions [BL16]

- MITM session hijacking
- Downgrading attacks
- …

flickr

>_ SSH

IPsec

SSL SECURED

And in the real world?

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Length Extension (e.g., SHA-1, SHA-2)

- Not a traditional property for cryptographic hashes
- From H(x) an adversary can produce H(x||y) without knowing x
- Example: Breaking authentication in Flickr

Transcript Collisions [BL16]

- MITM session hijacking
- Downgrading attacks
- …

flickr

>_ SSH

IPsec

SSL SECURED

Can we find these flaws **automatically** in the protocol design?

Automatic analysis & Hashes

Automatic analysis & Hashes

Existing Symbolic Model of Cryptography

- Automated security protocol analysis tools
- Assumes “perfect” hash functions (ROM)

Tamarin  , ProVerif

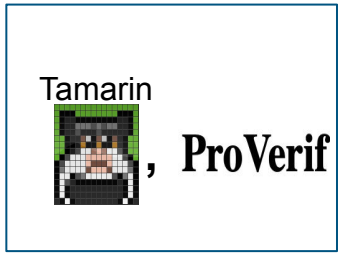
Automatic analysis & Hashes

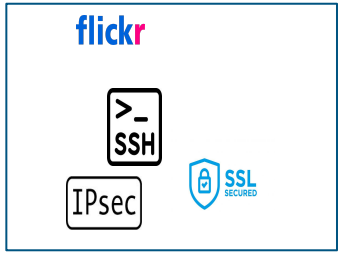
Existing Symbolic Model of Cryptography

- Automated security protocol analysis tools
- Assumes “perfect” hash functions (ROM)

Our Work: Find better Models

- Model all known hash weaknesses
- Goal: discover vulnerabilities in protocols automatically





Automatic analysis & Hashes

Existing Symbolic Model of Cryptography

- Automated security protocol analysis tools
- Assumes “perfect” hash functions (ROM)

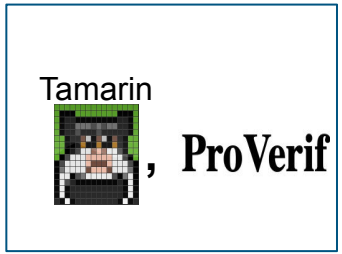
Our Work: Find better Models

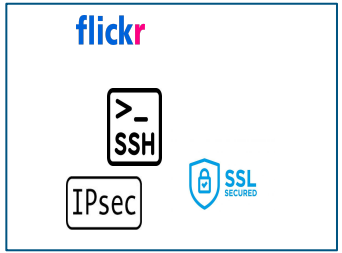
- Model all known hash weaknesses
- Goal: discover vulnerabilities in protocols automatically

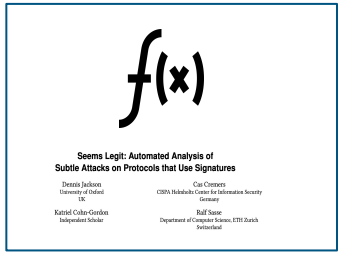
A Technical Detail: Non-Classical Modelling

Underspecified functions approach **[JCCS19]**

- Tool explores all possible functions that meet the requirements
- Trace restrictions limit the possible functions (e.g., forbid collisions)







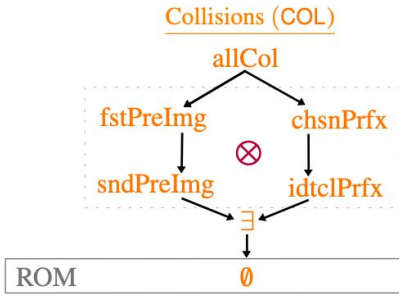
Symbolic hash models

We classify our attack models into 4 dimensions

Symbolic hash models

We classify our attack models into 4 dimensions

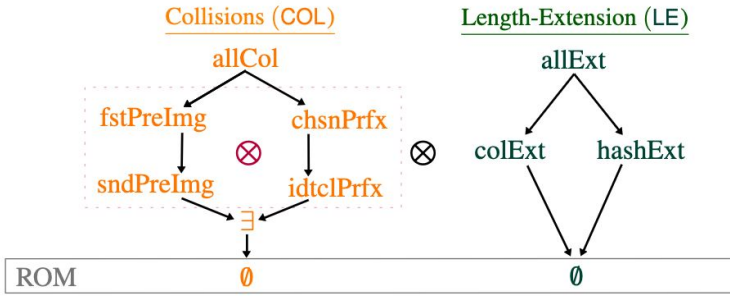
- Collisions



Symbolic hash models

We classify our attack models into 4 dimensions

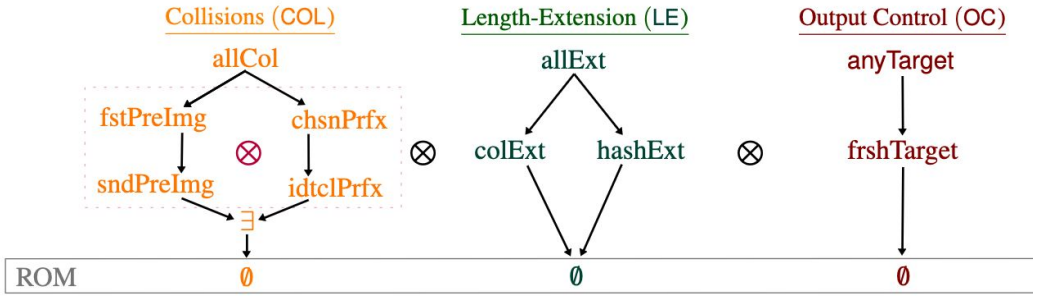
- Collisions
- Length Extensions



Symbolic hash models

We classify our attack models into 4 dimensions

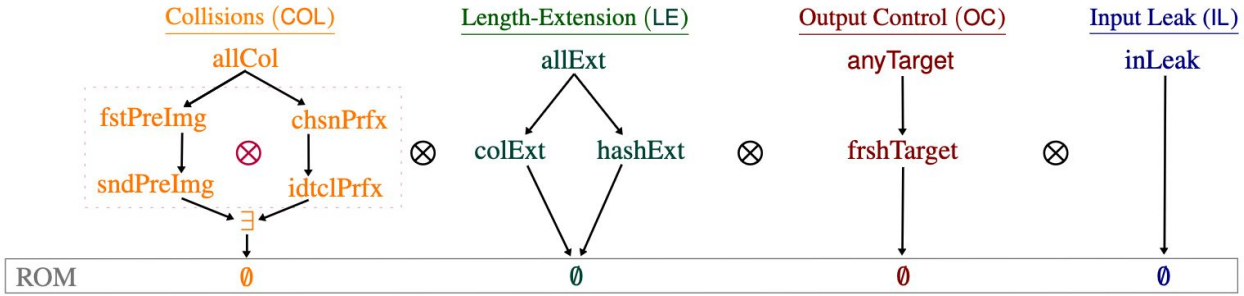
- Collisions
- Length Extensions
- Output Control



Symbolic hash models

We classify our attack models into 4 dimensions

- Collisions
- Length Extensions
- Output Control
- Leakage of input

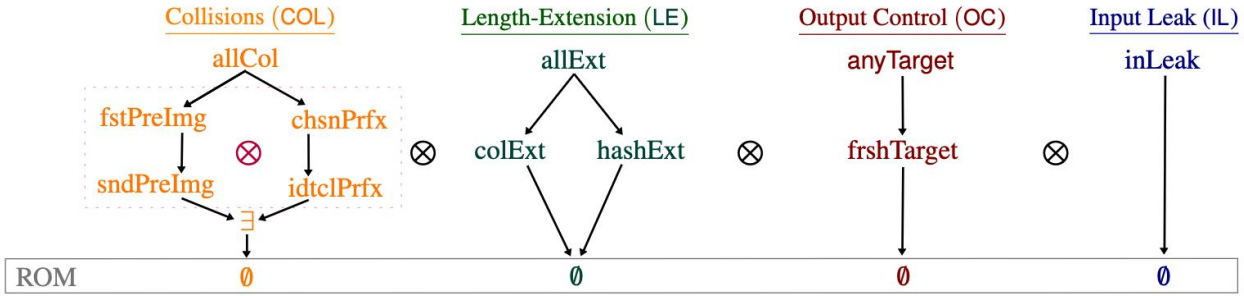


Symbolic hash models

We classify our attack models into 4 dimensions

- Collisions
- Length Extensions
- Output Control
- Leakage of input

One attack model consists of the combination of all dimensions

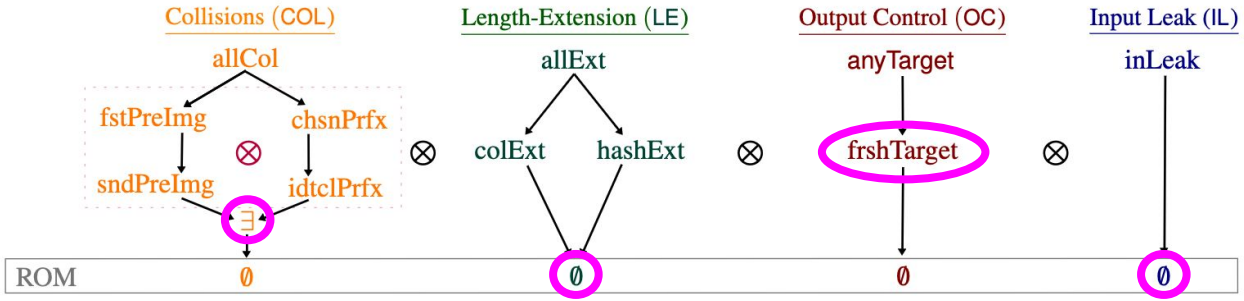


Symbolic hash models

We classify our attack models into 4 dimensions

- Collisions
- Length Extensions
- Output Control
- Leakage of input

One attack model consists of the combination of all dimensions



Finding a single collision between 2 random values

Symbolic hash models

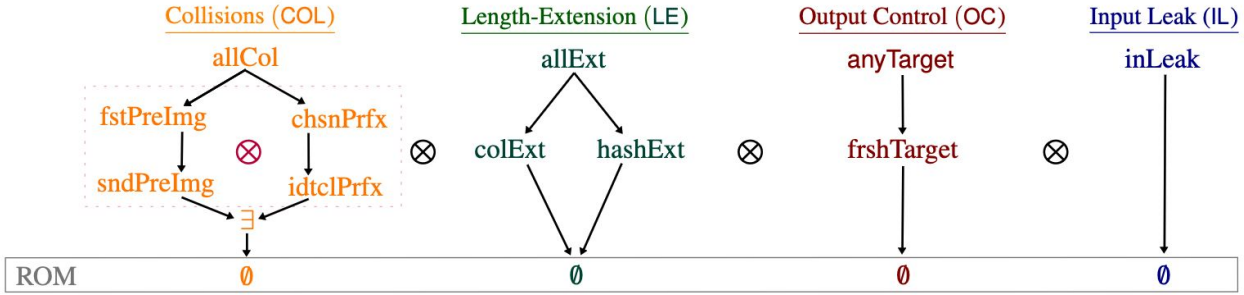
We classify our attack models into 4 dimensions

- Collisions
- Length Extensions
- Output Control
- Leakage of input

Tamarin

, ProVerif

One attack model consists of the combination of all dimensions



Symbolic hash models

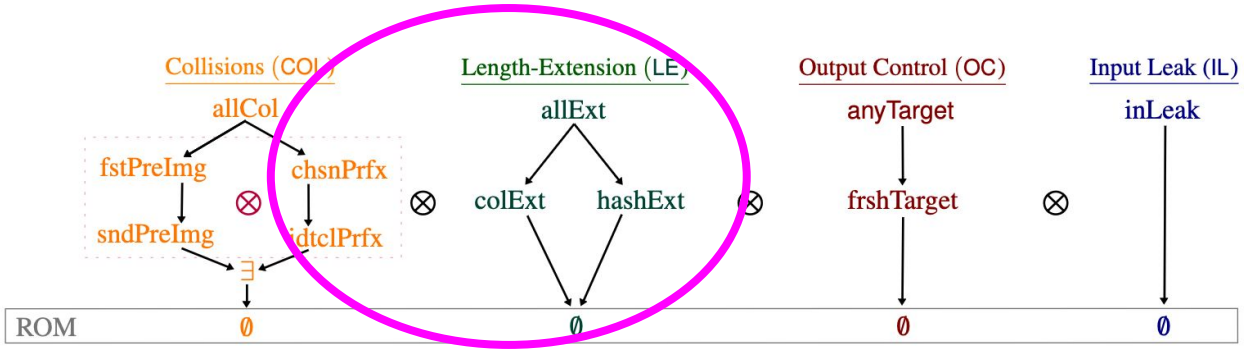
We classify our attack models into 4 dimensions

- Collisions
- Length Extensions
- Output Control
- Leakage of input

Tamarin

, ProVerif

One attack model consists of the combination of all dimensions



We need to extend the tools

Hashing lists and transcripts

Hashing lists and transcripts

Problems without associativity

Hashing lists and transcripts

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Problems without associativity

E.g., a list of 3 is expressed as a nested tuple

- <<a,b>,c>

Hashing lists and transcripts

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Problems without associativity

E.g., a list of 3 is expressed as a nested tuple

- <<a,b>,c>

Hashing a list of 3 could be either

- H(<<a,b>,c>) ≠ H(<a,<b,c>>)

Hashing lists and transcripts

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Problems without associativity

E.g., a list of 3 is expressed as a nested tuple

- <<a,b>,c>

Hashing a list of 3 could be either

- H(<<a,b>,c>) ≠ H(<a,<b,c>>)

Tamarin: New operator || ⇒ H(a||b||c)

Hashing lists and transcripts

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Problems without associativity

E.g., a list of 3 is expressed as a nested tuple

- <<a,b>,c>

Hashing a list of 3 could be either

- H(<<a,b>,c>) ≠ H(<a,<b,c>>)

Tamarin: New operator || ⇒ H(a||b||c)

Challenge

- No algorithm to guarantee a finite set of unifiers

Solution

- connected to underlying new maude feature

Hashing lists and transcripts

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Problems without associativity

E.g., a list of 3 is expressed as a nested tuple

- <<a,b>,c>

Hashing a list of 3 could be either

- H(<<a,b>,c>) ≠ H(<a,<b,c>>)

Tamarin: New operator || ⇒ H(a||b||c)

Challenge

- No algorithm to guarantee a finite set of unifiers

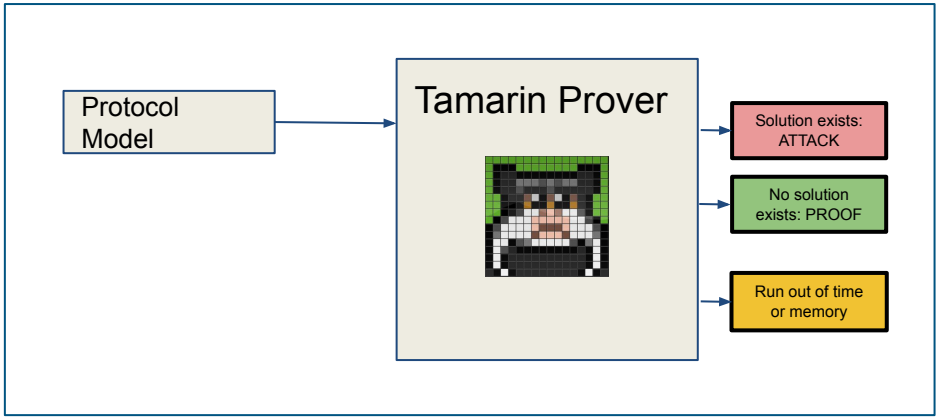
Solution

- connected to underlying new maude feature

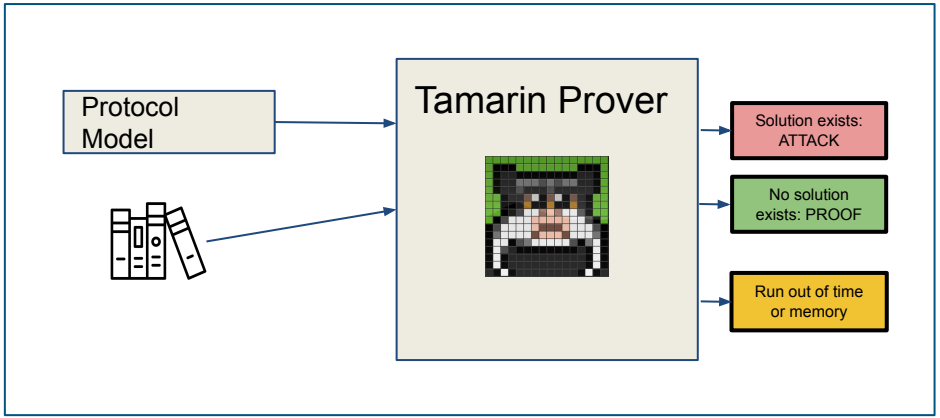
Proverif: Approximation

introduced recursive computation functions to define functions through general axiomatizations

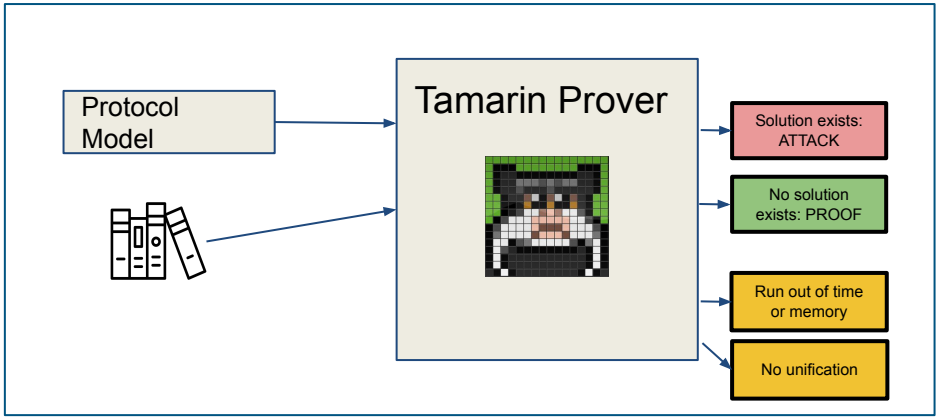
Systematic Analysis (focus on Tamarin)



Systematic Analysis (focus on Tamarin)



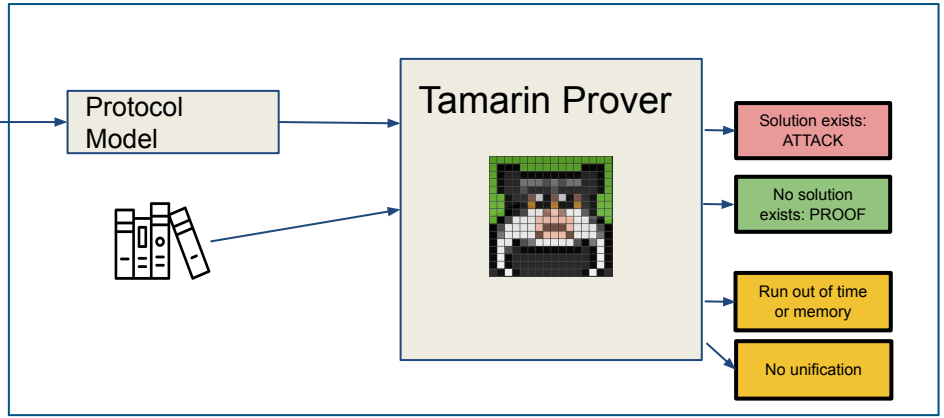
Systematic Analysis (focus on Tamarin)



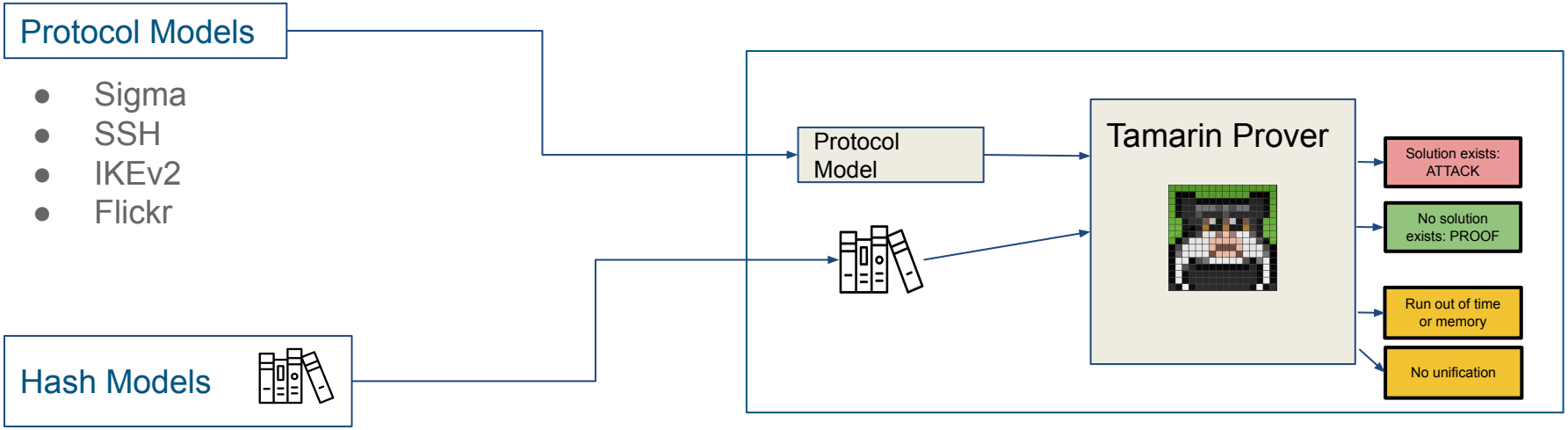
Systematic Analysis (focus on Tamarin)

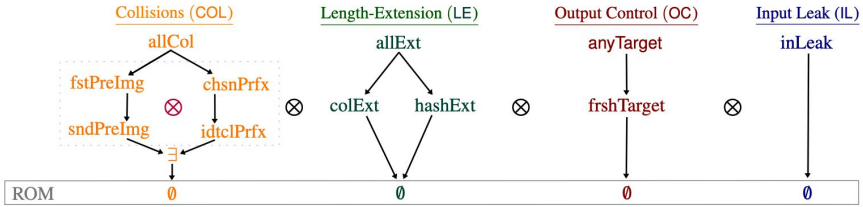
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr



Systematic Analysis (focus on Tamarin)





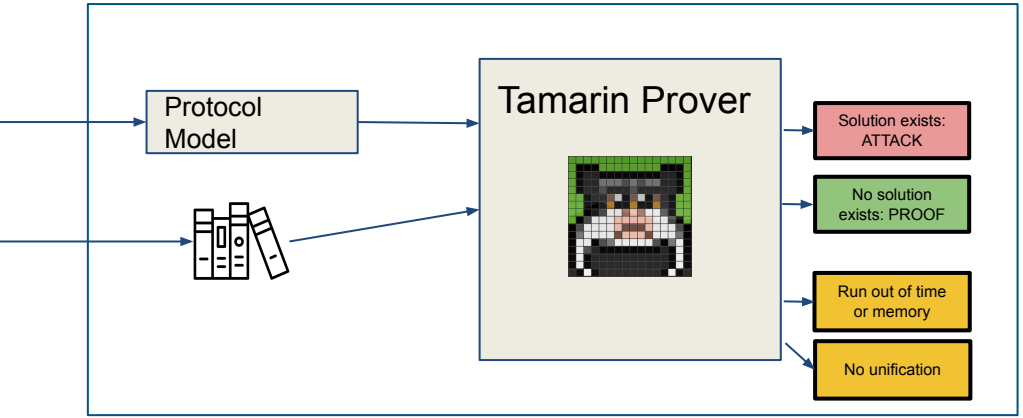
Systematic Analysis (focus on Tamarin)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

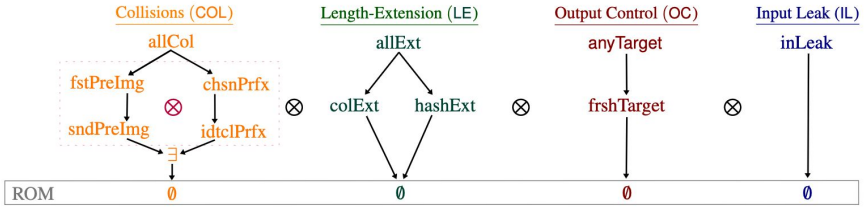
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr

Hash Models



Automatically construct all combinations



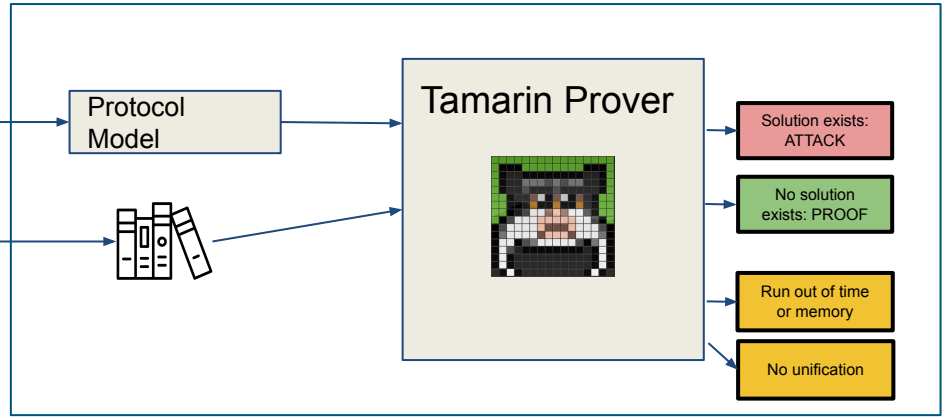
Systematic Analysis (focus on Tamarin)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

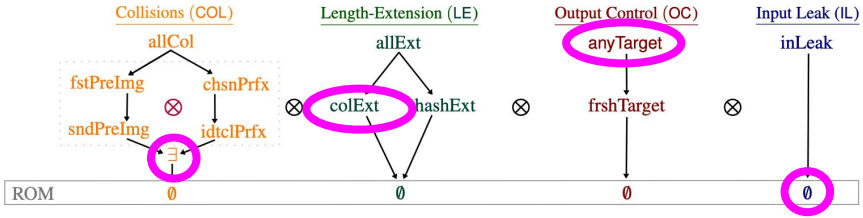
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr

Hash Models



Automatically construct all combinations



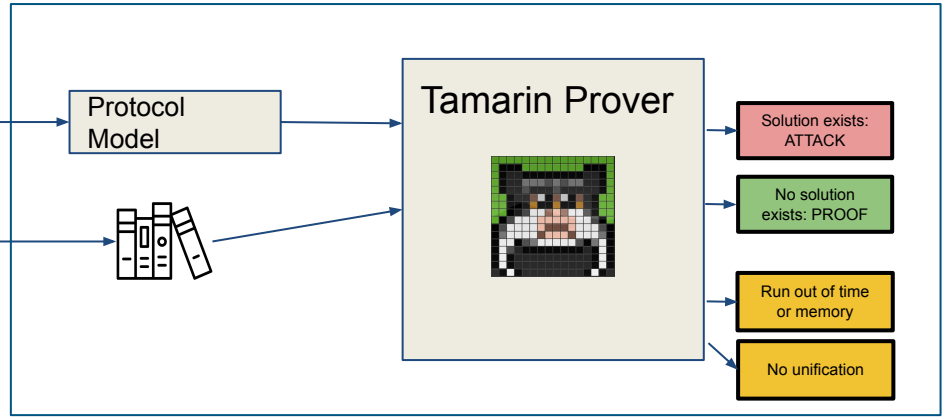
Systematic Analysis (focus on Tamarin)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

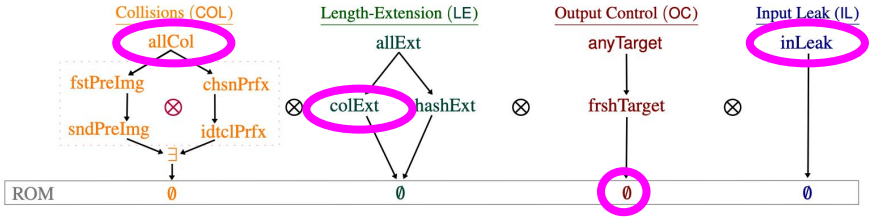
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr

Hash Models



Automatically construct all combinations

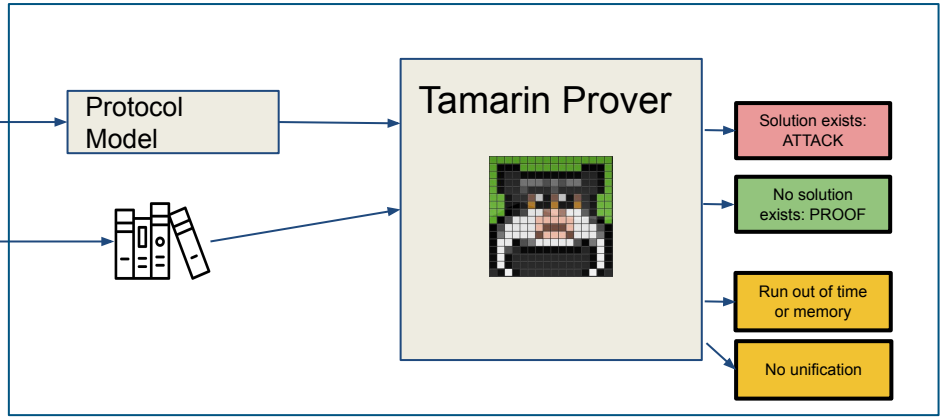


Systematic Analysis (focus on Tamarin)

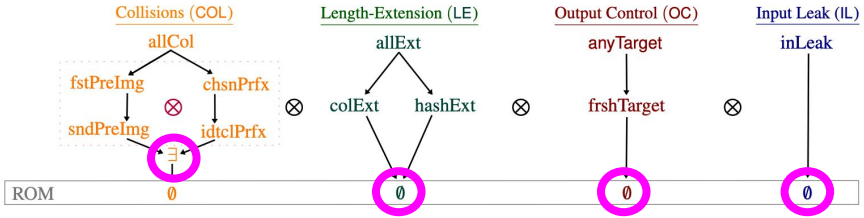
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr

Hash Models



Automatically construct all combinations



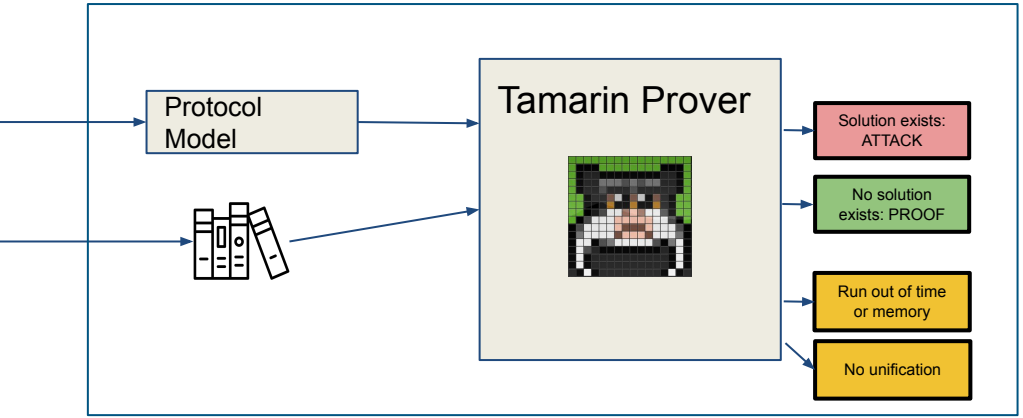
Systematic Analysis (focus on Tamarin)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

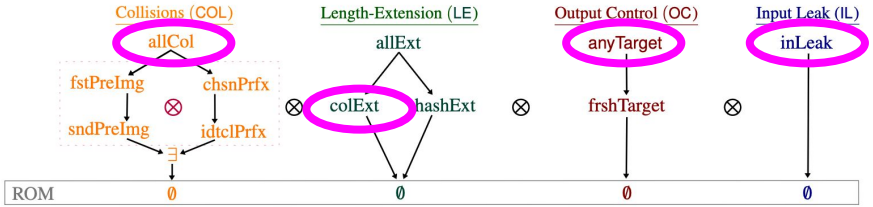
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr

Hash Models



Automatically construct all combinations

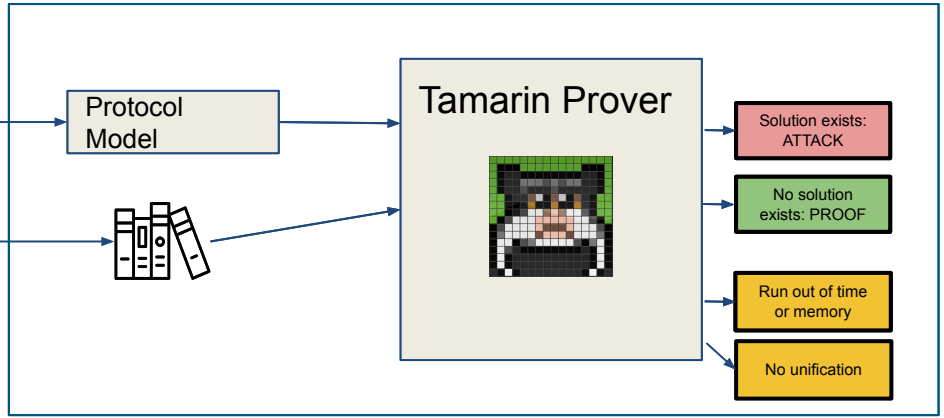


Systematic Analysis

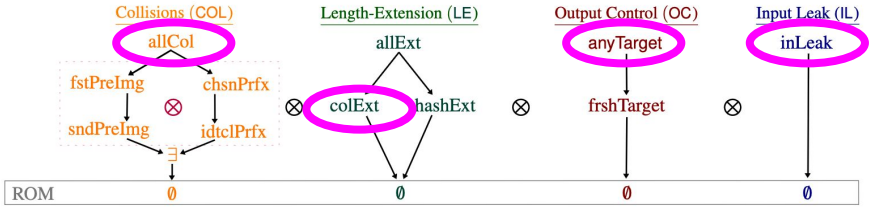
Protocol Models

- Sigma
- SSH
- IKEv2
- Flickr

Hash Models



Automatically construct all combinations



Tamarin, ProVerif

Implementations: https://github.com/charlie-j/symbolic-hash-models

Attacks automatically found

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Protocol	Main attack requirements	New?	Broken Property	Time(s)
Sigma	chsnPrfx,colExt	✗	Secrecy, Agreement	28
	chsnPrfx,colExt	✗ ~	Secrecy, Agreement	manual
	chsnPrfx	✓	Secrecy, Agreement	55
SSH	idtclPrfx,colExt	✗	Agreement	28
	sndPreImg,colExt	✓	Agreement	41
IKEv2	idtclPrfx,colExt	✗	Authentication	20
	∃ ,colExt	✓	Agreement	9
Flickr	hashExt	✗	Authentication	9

Attacks automatically found

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Protocol	Main attack requirements	New?	Broken Property	Time(s)
Sigma	chsnPrfx,colExt	✗	Secrecy, Agreement	28
	chsnPrfx,colExt	✗ ~	Secrecy, Agreement	manual
	chsnPrfx	✓	Secrecy, Agreement	55
SSH	idtclPrfx,colExt	✗	Agreement	28
	sndPreImg,colExt	✓	Agreement	41
IKEv2	idtclPrfx,colExt	✗	Authentication	20
	∃ ,colExt	✓	Agreement	9
Flickr	hashExt	✗	Authentication	9

Hash Gone Bad:

Automated discovery of protocol attacks that exploit hash function weaknesses

First automated methodology to find a large class of attacks

- Built new symbolic models for hash functions
- We extended both ProVerif and Tamarin
- Applied to several case studies, automatically finding attacks

Thanks for the Distinguished Paper Award!

Alexander Dax: alexander.dax@cispa.de

Artifact: https://github.com/charlie-j/symbolic-hash-models

Paper: https://www.usenix.org/conference/usenixsecurity23/presentation/cheval

References

[BL16] Bhargavan, K., & Leurent, G. (2016, February). Transcript collision attacks: Breaking authentication in TLS, IKE, and SSH. In *Network and Distributed System Security Symposium--NDSS 2016*.

[JCCS19] Jackson, D., Cremers, C., Cohn-Gordon, K., & Sasse, R. (2019, November). Seems legit: Automated analysis of subtle attacks on protocols that use signatures. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2165-2180).