




# Towards nonce-misuse resistant protocols

Tristan Claverie*, joint work with Gildas Avoine, Stéphanie Delaune

*: Agence nationale de la sécurité des systèmes d'information

Apr 05, 2024

# Incorrect nonce handling : protocol implementations

- WPS [1] : predictable nonces => authentication bypass, session key leaks
- LoRaWAN 1.0 [2] : reused nonces => replay, messages confidentiality weakens
- Bluetooth secure numeric comparison [3] : predictable nonces => authentication bypass, session key leaks

1. ***Offline bruteforce attack on WiFi Protected Setup***, **D. Bongard, Hack.Lu '14**
2. ***Rescuing LoRaWAN 1.0***, **G. Avoine and L. Ferreira, FC '18**
3. ***Bluetooth Randomness is Mostly Random***, **J. Tillmanns, J. Classen, F. Rohrbach, and M.Hollick, WOOT '20**

# Incorrect nonce handling : symmetric cryptography

- Misuse-resistance defined in 2006 for authenticated encryption [4]
- Some authenticated encryption modes have been designed to be misuse-resistant

4. ***A Provable-Security Treatment of the Key-Wrap Problem***, **P. Rogaway and T. Shrimpton, EUROCRYPT '06**

# Incorrect nonce handling : symbolic models

Nonces are perfect ?

# Goals

Using symbolic models :

- Incorrect implementation of nonces : what impact ?
- Are there more protocols more resilient than others ?

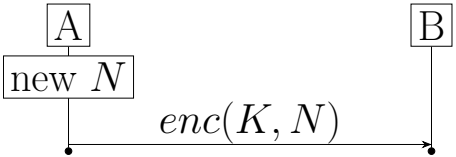
# Outline

# 1. Tackling nonce misuse in symbolic models

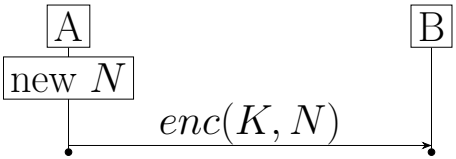
# Nonce properties

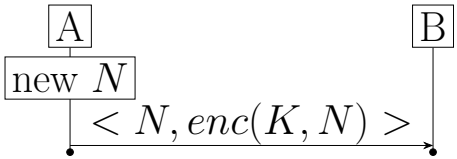
- Freshness
- Unpredictability

# Modelling nonce predictability



# Modelling nonce predictability

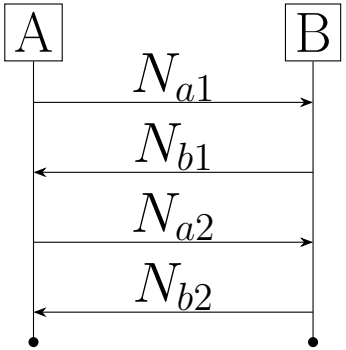


# Modelling reuse 1/2

Which nonce is considered to be reused ?
when ?

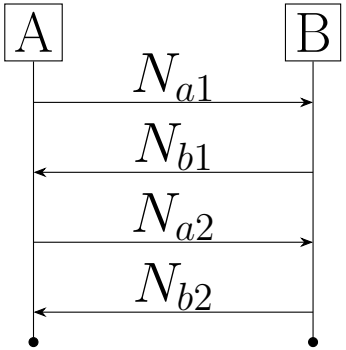
Example protocol :



# Modelling reuse 1/2

Which nonce is considered to be reused ? when ?

Example protocol :

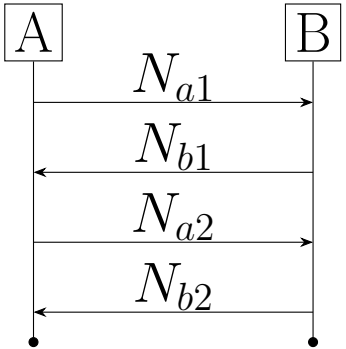



Deriving nonce reuse cases :

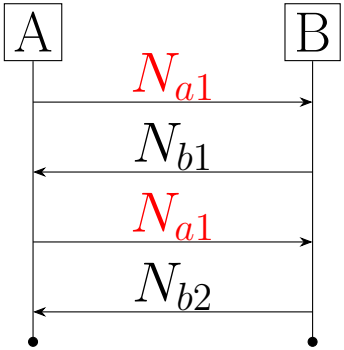
# Modelling reuse 1/2

Which nonce is considered to be reused ? when ?

Example protocol :



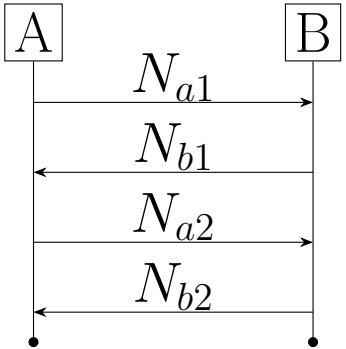

Deriving nonce reuse cases :



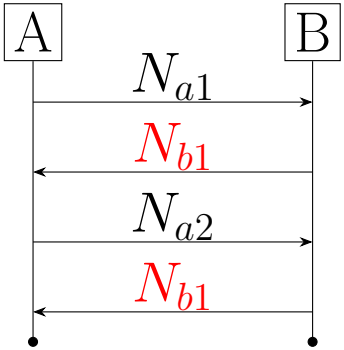
# Modelling reuse 1/2

Which nonce is considered to be reused ? when ?

Example protocol :



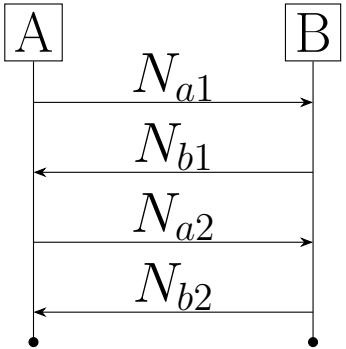

Deriving nonce reuse cases :



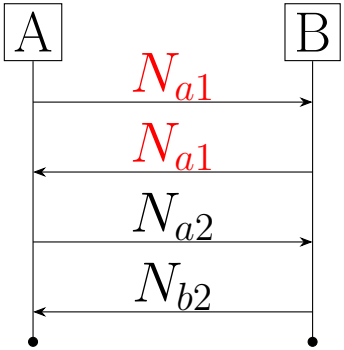
# Modelling reuse 1/2

Which nonce is considered to be reused ? when ?

Example protocol :



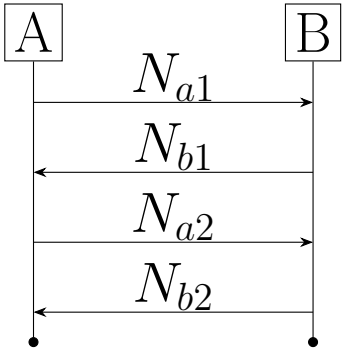

Deriving nonce reuse cases :



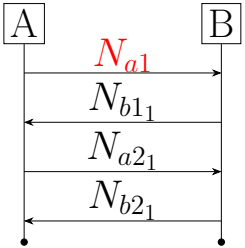
# Modelling reuse 1/2

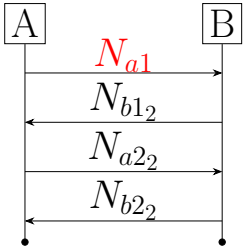
Which nonce is considered to be reused ? when ?

Example protocol :




Deriving nonce reuse cases :



# Reuse 2/2

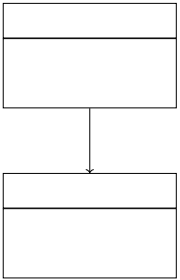
Considering a protocol between two agents, two roles, for an unbounded number of sessions :

- Same agent, same role, same session ;
- Same agent, same role, different session ;
- Same agent, different role, different session ;
- Different agent, different role, same session ;
- ...

# 2. A proposed Tamarin representation

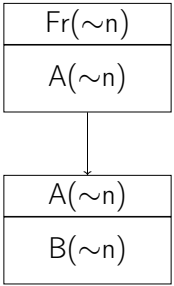
# Nonce reuse in Tamarin

Idea : (Mis)generate nonces in a specific rule



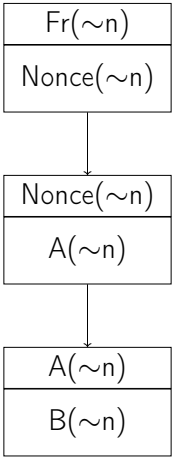
# Nonce reuse in Tamarin

Idea : (Mis)generate nonces in a specific rule



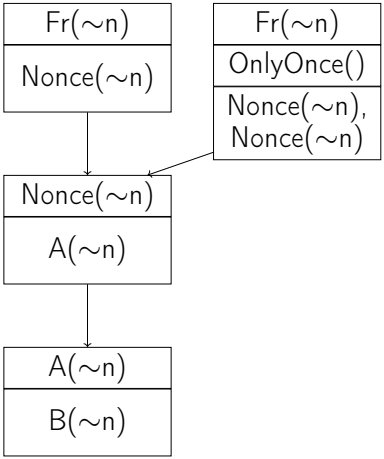
# Nonce reuse in Tamarin

Idea : (Mis)generate nonces in a specific rule



# Nonce reuse in Tamarin

Idea : (Mis)generate nonces in a specific rule



# Implementation choices

Option 1 : $\texttt{Nonce}(\sim n)$

- The same nonce may be used twice for different agents/roles/sessions
- + If security properties are proven, any kind of nonce reuse has no impact
- - If they are not, which scenarios hold and which do not ?

# Implementation choices

Option 1 : $\texttt{Nonce}(\sim n)$

- The same nonce may be used twice for different agents/roles/sessions
- + If security properties are proven, any kind of nonce reuse has no impact
- - If they are not, which scenarios hold and which do not ?

Option 2 (outline) : $\texttt{Nonce}('id', \sim n)$

- Add identifiers to Nonce(...) facts
- Those identifier allow to misgenerate nonces only for a single agent/role/session

# Pros and cons

Modelling reuse :

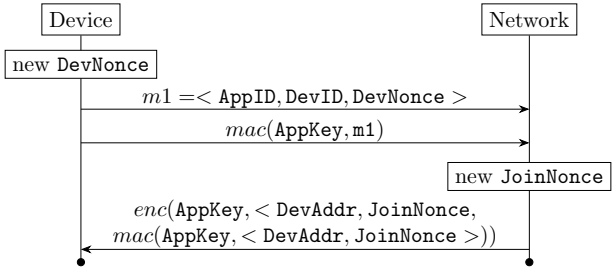
- Multiple option possibles, some alternatives may exist
- Modelling choice depends on what is studied :
  - Option 1 : Easy to modify 'Fr' with 'Nonce', hard to understand which scenario fails
  - Option 2 : More difficult to modify the protocol, easy to understand the case studied

# 3. Implementation and results

# Generating misuse cases

- We introduce the special Tamarin fact "Nonce"
- Implement a Tamarin parser to get the Abstract Syntax Tree
- Modify the AST to generate all possible misuse cases
- Output models are analysed separately with Tamarin.

# Case study : LoRaWAN 1.0




$\texttt{AppSKey} = enc(\texttt{AppKey}, <' 1', \texttt{JoinNonce}, \texttt{DevNonce} >)$
$\texttt{NwkSKey} = enc(\texttt{AppKey}, <' 2', \texttt{JoinNonce}, \texttt{DevNonce} >)$

# Generated cases :

- Basic case : nonces are fresh and unpredictable
- 2 leak cases : DevNonce and JoinNonce leak as soon as they are generated
- 4 reuse cases :
  - Device reuses DevNonce in two sessions
  - Network reuses JoinNonce in two sessions
  - Device always uses the same DevNonce
  - Network always uses the same JoinNonce

# Results :

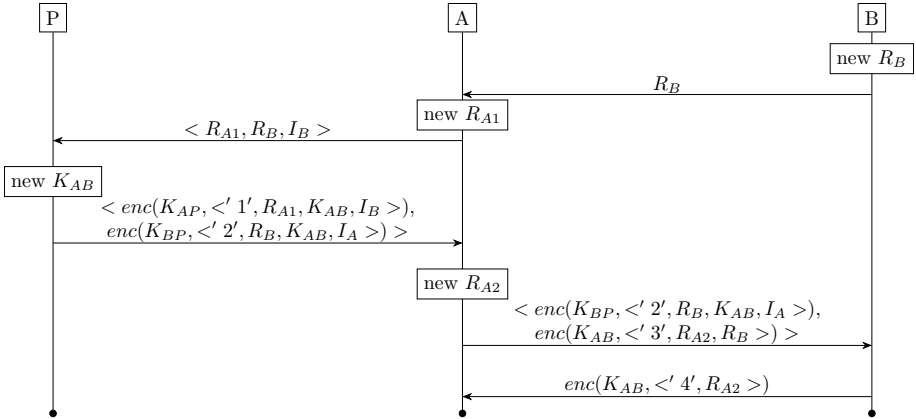
| Case | Agr D | Agr N | Inj Agr D | Inj Agr N | Fresh Session Key N | Fresh Session Key D |
|---|---|---|---|---|---|---|
| Basic | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| Leak D | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| Leak N | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| Reuse Once D | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Reuse Once N | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Reuse Always D | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Reuse Always N | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |

# More case studies :

ISO 9798 key agreement protocols :

- 9798-2-1 : injective agreement fails on both sides when nonce reused
- 9798-2-4 : injective agreement fails on both sides when nonce reused
- 9798-2-6 : injective agreement fails on one side when nonce reused
- 9798-3-4 : injective agreement fails on both sides when nonce reused
- 9798-4-4 : injective agreement fails on both sideswhen nonce reused

# ISO 9798-2-6


P
A
B
new $R_B$
$R_B$
new $R_{A1}$
$< R_{A1}, R_B, I_B >$
new $K_{AB}$
$< enc(K_{AP}, <' 1', R_{A1}, K_{AB}, I_B >),$
$enc(K_{BP}, <' 2', R_B, K_{AB}, I_A >) >$
new $R_{A2}$
$< enc(K_{BP}, <' 2', R_B, K_{AB}, I_A >),$
$enc(K_{AB}, <' 3', R_{A2}, R_B >) >$
$enc(K_{AB}, <' 4', R_{A2} >)$

# 4. Conclusion

# Conclusion

- Identify two core properties for nonces, that are not always granted by implementations
- Define ways to model misimplementation of those properties in Tamarin
- Study some concrete protocols with the modifications made

# Future work

- More complex case studies
- Identify additional possible misuses
- Find resistant protocols ?