A Formally Verified Device Authentication Protocol Using Casper/FDR

Mahdi Aiash, Glenford Mapp*, Raphael C.-W. Phan†, Aboubaker Lasebae, Jonathan Loo*
*School of Engineering and Information Science
Middlesex University, London, UK
Email: M.Aiash, G.Mapp, A.Lasebae, J.Loo@mdx.ac.uk
†Electronic and Electrical Engineering
Loughborough University, Loughborough, UK
Email: R.Phan@lboro.ac.uk

***Abstract*—For communication in Next Generation Networks, highly-developed mobile devices will enable users to store and manage a lot of credentials on their terminals. Furthermore, these terminals will represent and act on behalf of users when accessing different networks and connecting to a wide variety of services. In this situation, it is essential for users to trust their terminals and for all transactions using them to be secure. This paper analyses a number of the Authentication and Key Agreement protocols between the users and mobile terminals, then proposes a novel device authentication protocol. The proposed protocol is analysed and verified using a formal methods approach based on Casper/FDR compiler.**

***Keywords*-Authentication and Key Agreement Protocol; Device Authentication Protocols; Casper/FDR;**

I. Introduction

Due to the high popularization of wireless and cellular technologies such as WLAN, WiMAX and 2.5G/ 3G networks, an increasing number of services will be available in the mobile environment. Examples of these services are e-Commerce, on line banking and electronic public services in addition to access to email, Grid and Cloud resources/services. End users will access these services using mobile devices such as smart phones, PDAs and laptops. These devices will act on behalf of the users and hence may retain sensitive and credential information such as passwords, security certificates, secret keys and subscription IDs which is used to access the mobile services. This situation highlights the need for securing transactions between the end users and their mobile devices as well as the need for maintaining the integrity of the mobile devices. Creating such a secure environment will emphasise on the trustworthiness of mobile devices and encourage end users to delegate their devices the communication with sensitive services.

Therefore, this paper analyses some Authentication and Key Agreement (AKA) protocols in the literature such as the AKA protocol in [2] and the AKA protocol of Mobile Ethernet [3]. Based on this analysis, a two-stage AKA protocol is introduced. In the first stage, the protocol achieves mutual authentication and sets up a secure connection between the mobile device and the Personal Identification Card (PIC) which, similar to the Subscriber Identity Module (SIM) and Universal Subscriber Identity Module (USIM) cards in GSM and UMTS, holds security and subscription information. The second stage of the protocol authenticates the user to use the mobile device based on biometric information, and thus mutual authentication between the Mobile device, the PIC and the end user is achieved. Furthermore, the proposed protocol is authenticated using CASPER/FDR compiler [4] which is a formal methods-based approach that accepts an abstract description of systems and translates them into Communication Sequential Processes (CSP) [5], the generated CSP description is then verified using the the Failure Divergence Refinement (FDR) model checker [6].

The contributions of this work are as follows: Firstly, using Casper/FDR, we formally model some of the initial AKA protocols such as the one proposed by the Mobile Ethernet [3] and analyse the discovered attack. Secondly, to address the discovered drawbacks of the protocol in the literature, a new AKA protocol is introduced. The proposed protocol is modelled using Casper/FDR and then analysed against a number of desired security properties.

The rest of paper is organized as follows: Section 2 describes related work in the literature. The proposed AKA protocol is introduced and analysed in Section 3. The paper concludes in Section 4.

II. Related Work

A. Verifying Security Protocols Using Casper/FDR

Previously, analysing security protocols used to be done using two stages. Firstly, modelling the protocol using a theoretical notation or language such as the CSP [5]. Secondly, verifying the protocol using a model checker such as Failures-Divergence Refinement (FDR) [6].

However, describing a system or a protocol using CSP is a quite difficult and error-prone task; therefore, Gavin Lowe [4] has developed the CASPER/FDR tool to model security protocols, it accepts a simple and human-friendly input file that describes the system and compiles it into CSP code which is then checked using the FDR model checker. Casper/FDR has been used to model communication and security protocols as in [7], [8], [9]. The CASPER's input

Table I
THE HEADERS OF CASPER'S INPUT FILE

The Header	Description
# Free Variables	Defines the agents, variables and functions in the protocol
# Processes	Represents each agent as a process
# Protocol Description	Shows all the messages exchanged between the agents
# Specification	Specifies the security properties to be checked
# Actual Variables	Defines the real variables, in the actual system to be checked
# Functions	Defines all the functions used in the protocol
# System	Lists the agents participating in the actual system with their parameters instantiated
# Intruder Information	Specifies the intruder's knowledge and capabilities

file that describes the systems consists of eight headers as explained in Table I:

B. Desired Security Features for AKA protocols

As stated in [10], it is desired for AKA protocols to meet certain security properties. Therefore, a list of these properties will be used to analyse the proposed AKA protocol in this paper.

1) *Mutual Entity Authentication*: This is achieved when each party is assured of the identity of the other party.
2) *Mutual Key Authentication*: This is achieved when each party is assured that no other party aside from a specifically identified second party gains access to a particular secret key.
3) *Mutual Key Confirmation*: This requirement means that each party should be assured that the other has possession of a particular secret key.
4) *Key Freshness*: a key is considered fresh if it can be guaranteed to be new and not reused through actions of either an adversary or authorized party.
5) *Unknown-Key Share*: In this attack the two parties compute the same session key but have different views of their peers in the key exchange. In other words, in this attack an entity A ends up believing that it shares a key with B; although this is the case, B mistakenly believes the key is instead shared with an entity $E \neq A$.
6) *Key Compromise Impersonation Resilience*: This property implies that if the Intruder compromised the long-term key of one party, he should not be able to masquerade to the party as a different party.

C. AKA and Authorization Scheme

In [2], the authors propose an AKA and Authorization framework for 4G networks. At the initial stage, the framework combines password, Biometric information as well as public key infrastructure (PKI) to achieve mutual authentication between the user, the SIM card and the device. Based on the result of the authentication in the initial stage, the framework achieves authentication between the mobile device and the network.

Although it is stated in [2] that the framework was proven to be scalable and provides some desired security features such as multi-prong mutual authentication, the framework suffers from two major drawbacks: firstly, in order to provide a considerably robust platform for user access to sensitive services and data and achieve the authentication process in the initial stage, the framework associates the Trusted Computing (TC) with the PKI by implementing Trusted Mobile Platform (TMP) [11]. These represent major modifications to the architecture of mobile devices. Secondly, some of the required functions to deal with the PKI-complexity and checking the integrity of the mobile terminal do not consider the limitations of battery and processing power in small devices such as Mobile terminals and Personal Digital Assistant (PDAs). These two reasons make the framework inapplicable with current architecture and capabilities of mobile devices.

D. The Device Authentication Protocol of the Mobile Ethernet Security Framework

The Mobile Ethernet group has in [3] proposed an AKA framework that deals with security at the network and service levels as well as achieving mutual authentication between the user, SIM card (here referred to as The Personal Identification Card (PIC)) and the mobile terminal. The security system comprises the following entities:

- The Personal Identification Card (PIC): Similarly to the SIM card in 2, 2.5 and 3G technologies [1], the PIC holds user's credentials such as the subscribed services' IDs and security keys.
- The Mobile Terminal (MT): Is the user's device.

The solution proposed by the Mobile Ethernet Group comprises two-stage authentication protocol; the first stage is used in the initial authentication; when the PIC is plugged into the MT for the first time. This stage is based on PKI and it aims at achieving a mutual authentication between the MT and PIC and agreeing on a secret key (K) which will be stored in the PIC and the MT. After the initial authentication, a simplified protocol, based on the derived secret key (K), is used for any subsequent authentication process.

Similar to the case of the previous AKA in subsection II-C, due to its complexity, the author believes that PKI is not suitable for small devices. Furthermore, as will be explained in the following subsection, the formal verification results show attack against the authentication protocol in the second stage.

1) Analysing the Device Authentication Protocol of the Mobile Ethernet: As described in [3], after running the initial authentication protocol, the PIC and the MT will agree on a secret key (K), which will facilitate the subsequent authentications. By considering the notation in Table II, the authentication protocol runs as follows:

Table II
NOTATION.

Abbreviation	Full name and description
PIC	The Personal Identification (PIC), initially shares SK(MT) with the MT and holds the (UK)
MT	Mobile Terminal
r1, r2	Random numbers
K	A pre-shared secret key between the MT and the PIC
Req	An authentication request message
$MAC\{m\}_K$	Message authentication code of message (m) using the key (K)

The mobile terminal sends an authentication request (Req) to the PIC, which responds by sending a random value (R1) as a challenge towards the MT. The MT returns the hash of the R1 as well as a challenge (R2). The PIC responds to this challenge by sending the hashed R2.

$Msg1.MT \rightarrow PIC : Req$
$Msg2.PIC \rightarrow MT : R1$
$Msg3.MT \rightarrow PIC : MAC\{R1\}_K, R2$
$Msg4.PIC \rightarrow MT : MAC\{R2\}_K$

The full Casper's description of the protocol is mentioned in Appendix IV-A. After modelling this protocol, Casper/FDR discovered the following attack.

1a. MT -> I_PIC : req
1b. I_PIC -> PIC : req
2a. PIC -> I_MT : R1
2b. I_PIC -> MT : R1
3a. MT -> I_PIC : R2, MAC $\{R1\}_K$
3b. I_PIC -> PIC : R2, MAC $\{R1\}_K$
4a. PIC -> I_MT : MAC $\{R2\}_K$
4b. I_MT -> MT : MAC $\{R2\}_K$

The notation I_PIC, I_MT represents the case where the Intruder impersonates the PIC, MT respectively. As shown in Fig 1, the discovered attack is a Man-in-the-Middle attack, where the Intruder intercepts and replays the messages. This attack could be interpreted as follows: the mobile terminal will complete running the protocol believing that it was with the PIC, while it was with the Intruder instead. Similarly, the PIC will believe it has been running the protocol with the MT, while in reality it was with the Intruder.

III. The Proposed User-Level AKA Protocol

The proposed protocol comprises two stages: the first achieves mutual authentication between the PIC and the MT. ,while in the second, the user is authenticated based on his Biometric information. Similar to the AKA protocols in GSM and UMTS [1] and to avoid the overhead of PKI, the proposed protocol is based on a symmetric encryption

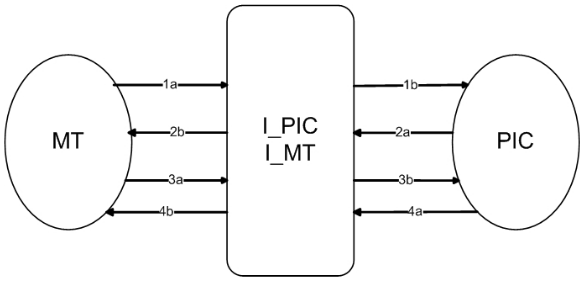


Figure 1. The Discovered Attack

Table III
NOTATION.

Abbreviation	Full name and description
PIC	The Personal Identification (PIC), initially shares SK(MT) with the MT
MT	Mobile Terminal
r1, r2, r3	Random numbers
miD	Mobile device unique ID
PSeq	PIC unique serial number
F, F1	Irreversible functions
K	A secret key, derived to secure the connection between the MT and the PIC: *K = F (SK(MT), r1, r2, miD, PSeq)*
SK(MT)	A pre-shared key between the PIC and the MT
Ackm	An Authentication Token: *Ackm = F1(MiD, PSeq, random)*
$Enc\{m\}_K$	Encrypting the message (m) using the key (K)

By considering the notations in Table III, the protocol runs as follows:

$$Msg1.PIC \rightarrow MT : \{r1, Pseq\}_{SK(MT)}$$

Upon plugging the Personal Identification Card PIC into the Mobile terminal MT, the AKA process starts by sending a random number r1 in Msg1.

$$Msg2.MT \rightarrow PIC : \{MiD, r1, r2\}_{SK(MT)}$$

The MT constructs a challenge message Msg2 containing a Mobile ID, a fresh challenge random r2 and the received random r1, this message is encrypted by the pre-shared key SK(MT). Using the information included in Msg2, both ends generate a secret key *K= F (SK(MT), r1, r2, miD, PSeq)* to secure the connection between the ends, the uniqueness of the derived key is based on the freshness of nonce r1, r2 and the secrecy of the pre-shared key SK.

$$Msg3.PIC \rightarrow MT : \{r3, r2\}_K$$

The PIC responds to the challenge in Msg2 by constructing Msg3 which contains the received challenge random r2 and another challenge random number r3, this message is encrypted using the derived secret key K.

$$Msg4. MT \rightarrow PIC : \{r3, Ackm\}_K$$

The MT responds by sending the received challenge r3 along with the pre-shared acknowledgement string Ackm via Msg4. As shown in Table III, the Ackm is derived in a way to include the identities of the two parties (the MT and the PIC), also it includes a fresh random value to guarantee the freshness, this way possessing the Ackm will help in achieving entity authentication as will be described in section III-A1.

$$Msg5. PIC \rightarrow MT : \{Ackm\}_K$$

The SP verifies the included Ackm in Msg4 and composes Msg5. In the case of a successful authentication among the PIC, the MT and the user, the MT represents the PIC and the user in the following stages of the AKA framework.

A. *Formal Verification*

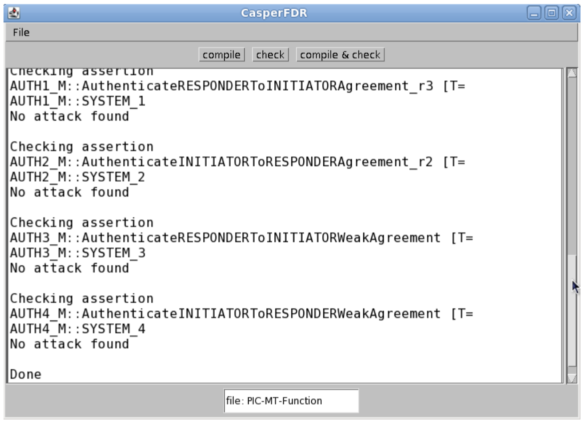
We modelled our protocol by preparing a Casper input file describing the UL-AKA protocol. For conciseness, we only show here the #Specification and #Intruder headings, while the #Free Variables, #Protocol Descriptions and #System headings are included in Appendix IV-B.

The #Free variables heading defines the participating parties, the variables and the used functions. It is worth noting that Casper does not specify a built-in method to simulate key derivation functions; therefore, we specifically defined therein the function F which is used to derive the session key (K) specific. The Protocol Description heading specifies how the intended parties will use the functions to generate the corresponding keys.

The security requirements of the system are defined under the #Specification heading. The lines starting with the keyword Secret define the secrecy properties of the protocol. For example, the first line specifies SK(MT) as a secret between the PIC and MT. The lines starting with Agreement define the protocol's authenticity properties; thus, the first authenticity of the figure above specifies that the MT is correctly authenticated to the PIC and agreed on the nonce value (r3). The WeakAgreement(X,Y) specification means that if Y thinks he has successfully completed a run of the protocol with X, then X has previously been running the protocol with Y.

# Specification

```
Secret(PIC,SK(MT),[MT])
Secret(MT,SK(MT),[PIC])
Secret(PIC,miD,[MT])
Secret(PIC,K,[MT])
Secret(MT,K,[PIC])
Agreement(MT,PIC,[r3])
Agreement(PIC,MT,[r2])
WeakAgreement(MT, PIC)
WeakAgreement(PIC, MT)
```



```
Checking assertion
AUTH1_M::AuthenticateRESPONDERToINITIATORAgreement_r3 [T=
AUTH1_M::SYSTEM_1
No attack found

Checking assertion
AUTH2_M::AuthenticateINITIATORToRESPONDERAgreement_r2 [T=
AUTH2_M::SYSTEM_2
No attack found

Checking assertion
AUTH3_M::AuthenticateRESPONDERToINITIATORWeakAgreement [T=
AUTH3_M::SYSTEM_3
No attack found

Checking assertion
AUTH4_M::AuthenticateINITIATORToRESPONDERWeakAgreement [T=
AUTH4_M::SYSTEM_4
No attack found

Done
```

Figure 2. Casper/FDR Verification Result

The #Intruder Information heading shows that the intruder identity is Mallory, the identities of all agents and the nonce R1 are included in the intruder initial knowledge. The Crackable keyword is used to simulate key compromise attack, where a key is compromised either through cryptographic techniques, or through the key being stolen and then used to lead to a failure of authentication in a subsequent session. We specify the pre-shared key SK as compromisable.

#Intruder Information

```
Intruder = Mallory
IntruderKnowledge = {PICard, Mobile, R1}
Crackable = presharedKeys
```

Running Casper/FDR tool verifies that none of the checked assertions defined in the #Specification heading was vulnerable to an attack as shown in Fig 2.

1) Protocol Analysis and Security Considerations: Although Casper/FDR has shown no attack against the proposed protocol, we need to carefully consider the result, Casper/FDR proves the protocol in the system specified in the System heading Appendix IV-B; however, the protocol might be vulnerable in another system. Further analysis of the protocol based on the security requirement list is given in this section.

1) Mutual Entity Authentication:
There is no direct specification within Casper to check this property, yet in order to show how our protocol could meet this requirement, we explicitly considered the Ackm value is generated as follows $Ackm = F(MiD, PSeq, random)$. This value is pre-stored in the PIC and Mobile terminal. In Msg 4, 5 each entity ensures the other party to have the

right Ackm, which includes the parties' identities as parameters, thus, enforcing entity authentication. If the MiD and Pseq were exposed, it is not feasible for the Intruder to generate the Ackm, because it does not know the right random value. Even if the Intruder recorded Msg5, it could not be used in next sessions because a fresh key K is used for each session.

2) Mutual Key Authentication:
   The mutual authentication between the MT and the PIC is based on the secrecy of the derived session key (K). We got Casper to check this using the Secret (PIC, K, [MT]) assertion check.
3) Mutual Key Confirmation:
   This requirement is achieved by performing the checks after Msg3 and 4 in the Protocol Description heading Appendix IV-B. By using the Decryptable function each party makes sure that the valid secret key K is possessed by the other part. If the any of the check failed, the protocol aborts.
4) Key Freshness:
   Casper does not have any function to check this requirement, so we included freshly generated values r1, r2 in the derivation function of the the session key K: $K = F(SK(MT), r1, r2, miD, PSeq)$; thus the fact that Casper does not detect any attack on the secrecy of the session key (K) implies that key freshness is not violated.
5) Unknown-Key Share:
   The Aliveness assertion is used to check this attack. Additionally, making a binding between the Keys and the parties' identity deals with this attack. This has been achieved in this protocol by including the identities of the MT and the PIC in the KDF of the K.
6) Key Compromise Impersonation Resilience:
   We modelled this requirement by specifying the long-term keys as crackable and using the Agreement assertion to check any breach of the authenticity feature.

B. Biometric Information-Based Authentication

For this stage, we assume that the Mobile terminal is equipped with a trusted Biometric information reader such as fingerprint reader. When the user makes the initial contract, a brief hashed value of the user's biometric-information is stored in the PIC.

After running the previous AKA protocol and setting up a secure channel between the MT and the PIC, the user is prompted to enter his biometric-information, the MT processes the data and generates a hashed value of the submitted info. This hashed value is passed to the PIC which compares it with the previously stored value. In case of match, the user is authenticated as the PIC owner and consequently to use the MT. From this point onwards, the MT will represent the user in both network and service level connections.

IV. Conclusions

In order for users to be able to delegate their devices to join various access networks and contact a huge number of services, credential information such as logins and crypto-parameters are stored on these devices. However, to facilitate such delegation, trust-relationship between the users and their devices has to be achieved. Therefore, this paper presented a novel AKA protocol to achieve mutual authentication and set up a secure connection between the user and its device. The protocol was verified using Casper/FDR tool, also it was analytically proven to meet certain desired security requirements.

References

[1] P. Chandra, *Bulletproof wireless security : GSM, UMTS, 802.11 and ad hoc security*. Newnes. pp. 129-158, Oxford, 2005

[2] Y. Zheng, D. He, X. Tang and H. Wang, *AKA and Authorization Scheme for 4G Mobile Networks Based on Trusted Mobile Platform*, In Proceedings of ICICIS, 2005.

[3] M. Kuroda, M. Inoue, A. Okubo, T. Sakakura, K. Shimizu, F. Adachi, *Scalable Mobile Ethernet and Fast Vertical handover*, In Proceedings of IEEE Wireless Communications and Networking Conference, 2004.

[4] G. Lowe, P. Broadfoot, C. Dilloway, *A compiler for the Analysis of security protocol. Version 1.12*,Oxford University Computing Laboratory, http://www.cs.ox.ac.uk/gavin.lowe/Security/Casper/manual.pdf, 2009.

[5] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe and A,W. Roscoe, *The modelling and analysis of security protocols*, PEARSON Ltd, 2010.

[6] Formal Systems LTD, *Failures-Divergence Refinement. FDR2 User Manual*,http://www.fsel.com/documentation/fdr2/fdr2manual.pdf, 2005.

[7] S. Xu, C. Tser Huang, M. Matthews, *Modelling and analysis of IEEE 802.16 PKM Protocols using CasperFDR*, In Proceedings of Wireless Communication Systems, ISWCS, 2008.

[8] K. Raju, V. Kumari, *Formal Verification of IEEE802.11i WPA-GPG Authentication Protocol*, In Proceedings of Communications in Computer and Information Science, 2011.

[9] M. Aiash, G. Mapp, A. Lasebae, R. Phan and J. Loo, *A Formally Verified AKA Protocol For Vertical Handover in Heterogeneous Environments using Casper/FDR*, EURASIP Journal on Wireless Communications and Networking 2012, 2012:57 doi:10.1186/1687-1499-2012-57, 2012. OpenSpringer.

[10] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[11] Y. Inamura, T. Nakayama, A. Takeshita, *Trusted Mobile Platform Technology for Secure Terminals*. NTT DoCoMo Technical Journal, 7, 25-39.

Appendix

A. The Device Authentication Protocol of the Mobile Ethernet

# Free Variables

```
PIC, MT : Agents
r1, r2, r3 : Nonce
miD : DeviceID
K : SessionKeys
h : HashFunction
Req: message
InverseKeys = (K, K)
```

# Pocesses

```
INITIATOR(PIC,r1, K )
RESPONDER(MT,PIC, r2, miD, K, Req)
```

# Protocol Description

```
0. -> PIC : MT
1. MT -> PIC : Req
2. PIC -> MT : r1
3. MT -> PIC : {r1}{K}%v, h(r1), r2
```

$[decryptable(v, K) andnth(decrypt(v, K), 1) == r1]$

```
4. PIC -> MT : {r2}{K}%w
```

$[decryptable(w, K) andnth(decrypt(w, K), 1) == r2]$

# Specification

```
Secret(PIC,K,[MT])
Secret(PIC,r2,[MT])
Secret(PIC,r1,[MT])
Agreement(MT,PIC,[r1, K])
Agreement(PIC,MT,[r2, K])
WeakAgreement(MT,PIC)
WeakAgreement(PIC,MT)
```

# Actual Variables

```
PICard, Mobile, Eve : Agents
R1,R2, R3 : Nonce
MID : DeviceID
k : SessionKeys
InverseKeys = (k, k)
req: message
```

# System

```
INITIATOR(PICard,R1, k)
RESPONDER(Mobile,PICard, R2, MID, k,
req)
```

# Intruder Information

```
Intruder = Mallory
IntruderKnowledge = {PICard, Mobile}
```

B. The Proposed AKA Protocol

# Free Variables

```
PIC, MT : Agents
r1, r2 : Nonces
r3 : challNonce
SK : Agents -> presharedKeys
F : presharedKeys x Nonces x Nonces x
DeviceID -> SessionKeys
miD : DeviceID
K : SessionKeys
h : HashFunction
Ackm: Acknolwedgment
InverseKeys = (K, K), (SK, SK),(F, F)
```

# Pocesses

```
INITIATOR(PIC,r1,r3,Ackm) knows SK(MT)
RESPONDER(MT,PIC, r2, miD, Ackm) knows
SK(MT)
```

# Protocol Description

```
0. -> PIC : MT
1. PIC -> MT : {r1}{SK(MT)}
```

$< K := F(SK(MT), r1, r2, miD) >$

```
2. MT -> PIC : {miD,r2,r1}{SK(MT)}
```

$< K := F(SK(MT), r1, r2, miD) >$

```
3. PIC -> MT : {r2,r3}{K}%v
```

$[decryptable(v, K) andnth(decrypt(v, K), 1) == r2]$

$< r3 := nth(decrypt(v, K), 2) >$

```
4. MT -> PIC : ({Ackm, r3}{K})%w
```

$[decryptable(w, K) andnth(decrypt(w, K), 1) == r3 andnth(decrypt(w, K), 2) == Ackm]$

```
5. PIC -> MT: {Ackm}{K}%w1
```

$[decryptable(w1, K) andnth(decrypt(w1, K), 1) == Ackm]$

# Specification

```
Secret(PIC,SK(MT),[MT])
Secret(MT,SK(MT),[PIC])
Secret(PIC,miD,[MT])
Secret(PIC,K,[MT])
Secret(MT,K,[PIC])
Agreement(MT,PIC,[r3])
Agreement(PIC,MT,[r2])
WeakAgreement(MT, PIC)
WeakAgreement(PIC, MT)
```

# Actual Variables

```
PICard, Mobile, Mallory : Agents
R1,R2: Nonces
R3, R4 : challNonce
MID : DeviceID
k : SessionKeys
InverseKeys = (k, k)
ACKM: Acknolwedgment
```

# Functions

```
symbolic SK, F
```

# System

```
INITIATOR(PICard,R1,R3, ACKM)
RESPONDER(Mobile,PICard, R2, MID, ACKM)
```

# Intruder Information

```
Intruder = Mallory
IntruderKnowledge = {PICard, Mobile,R1}
Crackable = presharedKeys
```