Research article

Performance and cryptographic evaluation of security protocols in distributed networks using applied pi calculus and Markov Chain

Ed Kamya Kiyemba Edris [a,*], Mahdi Aiash [b], Mohammad Ali Khoshkholghi [b], Ranesh Naha [c], Abdullahi Chowdhury [d], Jonathan Loo [e]

[a] School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, United Kingdom
[b] School of Science and Technology, Middlesex University, London, United Kingdom
[c] Centre for Smart Analytics, Federation University Australia, Melbourne, Australia
[d] School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, Australia
[e] School of Computer Engineering, University of West London, London, United Kingdom

ARTICLE INFO

Keywords:
Security protocols
Formal methods
Formal verification
Applied pi calculus
Performance evaluation
5G
Edge computing

ABSTRACT

The development of cryptographic protocols goes through two stages, namely, security verification and performance analysis. The verification of the protocol's security properties could be analytically achieved using threat modelling, or formally using formal methods and model checkers. The performance analysis could be mathematical or simulation-based. However, mathematical modelling is complicated and does not reflect the actual deployment environment of the protocol in the current state of the art. Simulation software provides scalability and can simulate complicated scenarios, however, there are times when it is not possible to use simulations due to a lack of support for new technologies or simulation scenarios. Therefore, this paper proposes a formal method and analytical model for evaluating the performance of security protocols using applied pi-calculus and Markov Chain processes. It interprets algebraic processes and associates cryptographic operatives with quantitative measures to estimate and evaluate cryptographic costs. With this approach, the protocols are presented as processes using applied pi-calculus, and their security properties are an approximate abstraction of protocol equivalence based on the verification from ProVerif and evaluated using analytical and simulation models for quantitative measures. The interpretation of the quantities is associated with process transitions, rates, and measures as a cost of using cryptographic primitives. This method supports users' input in analysing the protocol's activities and performance. As a proof of concept, we deploy this approach to assess the performance of security protocols designed to protect large-scale, 5G-based Device-to-Device communications. We also conducted a performance evaluation of the protocols based on analytical and network simulator results to compare the effectiveness of the proposed approach.

1. Introduction

Formal methods and automated tools can be used to achieve verification of cryptographic protocols by applying different approaches such as symbolic and computational modelling. These methods have been used to evaluate security properties for strong security guarantees in complex networks like Edge [1,2], Internet of Things (IOT) [3,4] and Fifth Generation Mobile Network (5G)'s

primary authentication [5–7], service security [8], secondary authentication [9] and cloud computing [10,11]. In the past, this led to multiple attacks being found in widely deployed mobile network protocols.

Other methods are being introduced to verify security protocols, such as timed interpreted systems to examine the time dependencies of the security protocols' executions [12]. Similarly, in [13], the behaviour of the security protocol is gathered including time parameters and various aspects of computer networks to check protocol vulnerability to attacks. In [14], a computational first-order logic that is sound with respect to quantum attackers is used for mechanical proofs of computational post-quantum security.

In addition, these verified protocols have to be evaluated for their performance to check overall effectiveness in terms of throughput, delay, and latency. Different approaches, such as real experiments, analytical, and simulation modelling can be used to evaluate and validate the performance of a security protocol of a communication system. Real experimental measurement requires the use of a test bed to conduct a performance evaluation of a protocol, but it is very expensive and complex. Simulation tools such as NS-3 [15] and OMNET++ [16] have been used to evaluate communication security protocols by building a network simulator to represent a computer network. Simulation provides scalability and can simulate complicated scenarios, however, there are times when it is not possible to use simulations due to a lack of tools, simulation plugins for the specific technology-simulation software need to be programmed to simulate new technologies such as 5G and the complexity of advanced communication systems. Additionally, analytical modelling based on a mathematical description of the protocols using applied mathematical theories such as queuing and probability can be used. To get an intuition about the protocol performance measurements, numerical methods are applied to the model using tools and analytical processes such as MATLAB [17] and stochastic process [18], respectively. It relies on factors assumptions, and theories that are translated into the model.

This paper uses a methodology that includes a protocol specification, formal verification, analytical and simulation modelling. The designing of security protocols requires formal verification, which can be achieved by using formal methods and automated tools that rely on *pi* calculus. We believe that analytical modelling is the best option in evaluating protocols designed and verified using formal methods. So protocols modelled and verified using ProVerif based on processes and applied calculus are used as a case study [19], therefore we believe analytical modelling based on the Markov Chain model [18] fits this purpose. This evaluation of a security protocol's performance relies on quantitative measures of cryptographic operations of a protocol based on the approach in [20]. The evaluation procedure uses special operational semantics of applied *pi* calculus [21], that enable us to use quantitative measures on processes describing cryptographic protocols by deriving Markov Chain. Every cryptographic operation has a cost on the system, which can be estimated through quantitative measures. The cost of cryptographic operation and exchange of message is specified and evaluated based on quantitative properties such as availability, speed and length [20]. The protocols are measured by describing them through a ProVerif process and with labelled semantics [21], associating a cost with each process's transition. This aspect goes beyond traditional qualitative evaluations and supports decision-making regarding protocol selection and optimization.

The transitions include enhanced labels that associate with cost [22]. This is achieved by assigning rates to transitions of system activities, whereby these rates reflect the architecture of the system model and the use of encryption schemes such as the Elliptic Curve Integrated Encryption Scheme (ECIES) and Public Key Infrastructure (PKI). The performance of the system is evaluated by mapping transitions with Markov Chain [20]. However, our work differs as we use applied *pi* calculus, ProVerif an automated proof verifier for security guarantees for specification, and simulation implementation of the protocols to compare the results. The quantitative measures and qualitative semantics are abstracted symbolically from ProVerif specification of protocols discussed in [6,8,9]. The work in this paper considers advance in computing and communication applications, whereby security protocol verification and evaluation are based on the association of quantitative measures to state transitions, processes, and algorithms when solving mathematical problems. To the best of our knowledge, the applied *pi* calculus and ProVerif verified protocol have not been quantitatively measured using the Markov Chain model based on semantics operations and bisimilarity labels.

In this paper, our contributions are summarized as follows:

- We explore how formal methods can support the evaluation of security protocol performance in relation to processes and cryptographic properties.
- We link formal verification methods with performance modelling techniques for an efficient performance evaluation process.
- We propose a formal method that evaluates the security protocol's cryptographic operatives and transitions driven by their specifications' semantics in ProVerif and applied *pi* calculus.
- We adopt a symbolic approach that uses enhanced operational semantics together with bisimilarity labels to associate rates with the applied *pi* calculus process as a communication system-based Markov Chain model.
- We quantify the cryptographic properties of protocols in terms of computational and communication costs and simulate them in NS-3 to evaluate their performance.
- We also categorize the results in terms of the efficiency, latency, and throughput of the protocols and compare the performances of different 5G security protocols using analytical and simulation analysis.

The rest of the paper is structured as follows. Section 2 discusses related work on performance evaluation and quantitative measurement methods. In Section 3, the protocols specifications and performance modelling techniques are defined. Section 4 presents the Continuous Time Markov Chain model and quantitative information derivation process. In Section 5, two protocols' performances are evaluated with mathematical methods. While in Section 6 the security protocols are simulated to evaluate performance simulation using analytical results. The paper concludes in Section 7.

Table 1
ProVerif: Informal semantics and syntax.

M, N ::=	Terms
x, y, z	Variable
a, b, c, k, s	Name
f(M1, . . . ,Mn)	Constructor application
D ::=	Expressions
M	Term
h(D1, . . . ,Dn)	Function application
fail	Failure
P, Q ::=	Processes
0	Nil
out(N, M); P	Output
in(N, x : T); P	Input
P | Q	Parallel composition
!P	!P replication
new a : T; P	Restriction
let x : T = D in P else Q	Expression evaluation
if M = N then P else Q	Conditional
event(M);P	Event
$\nu n.P$	Name restriction (“new”)
$N(x).P$	Message input
$\bar{N}\langle M\rangle.P$	Message input

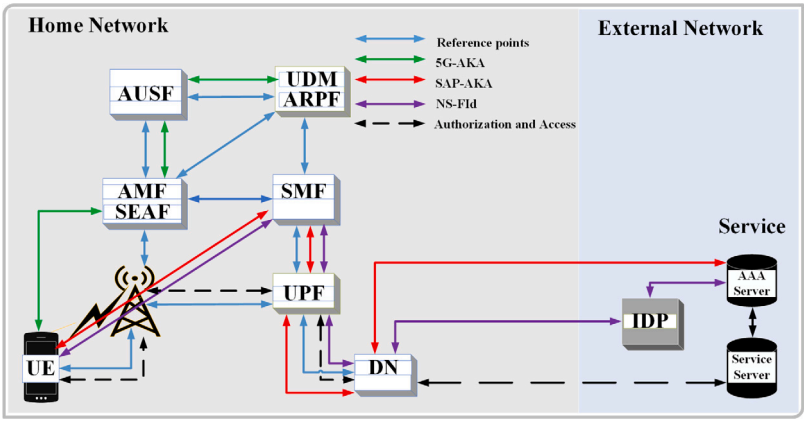


Fig. 1. 5G enhanced system model.

2. Related work

Many recent studies on 5G security have proposed security protocols, but few have evaluated the protocols’ performance [5–7,23]. Generally, performance evaluation is based on communication and computational overheads, while others rely on quantitative measurements from mathematical models. The authors in [24] proposed a secure authentication solution for 5G based on blockchain. They also evaluated the protocol performance concerning communication and computational overheads, with the result indicating that it is more efficient than the current schemes. In [25], the authors presented an efficient and secure 5G protocol for authentication and key agreement protocol, with the performance evaluation showing less communication and computational overhead.

The authors in [26] introduced an extension of *pi* calculus in the form of spi calculus, it was used to describe and analyse cryptographic protocols. The protocol is presented as a process in spi calculus, and protocol equivalence is used to state its security properties. In [21], the authors introduced an extension of *pi* calculus that provides primitive functions, equations, and value passing as terms based on informal semantics and proof approaches for reasoning about security protocols.

The authors in [20,22] used special operational semantics to describe cryptographic protocols and deduce quantitative measures. They also used system transitions, enhanced labels, and cost rates assigned to the transitions, reflecting the architecture applications, and different cryptosystems. The transition systems were mapped with the Markov Chain process and then evaluated the systems’ performance. In addition, a performance comparison was conducted of different processes and machine states with mobile computation to show the feasibility of their framework. In [27], the authors introduced methodologies that use Markov Chain processes for communication in networks using simple applications, cellular systems, and LAN models. These models were used to find steady-state distributions and compute the system performance metrics.

Table 2
SAP-AKA protocol specification.

Msg1.UE→SMF:({ServName, SID}, {KAMF})
Msg2.SMF→UE:({GPSI, SPID, PKSP}, {KAMF})
Msg3.UE→SPAAA:({ServName, SID}, {PKSP})
Msg4.SPAAA→UE:({Identity},{PKUE})
Msg5.UE→SPAAA:({GPSI},{PKSP})
Msg6.SPAAA→UE:({RAND, AUTN, MAC, KDF, KDF INPUT}, {KENC})
Msg7.UE→SPAAA:({AT RES, AT MAC2},{KENC})
Msg8.SPAAA→:({SUCCESS,KUE3A,EID},{KENC})
Msg9.SPAAA→UE:({SUCCESS, EID})

2.1. ProVerif and applied Pi calculus specifications

This subsection introduces the enhanced operation semantics, cost functions, and protocol specifications that are used to enhance transitions with their costs. It integrates the applied calculus process with enhanced labels.

Formal methods are mathematical model techniques used in the verification of systems by performing mathematical analysis. Applied *pi* calculus is a specification language that uses formal methods and notations [21], a widely used algebraic method for specifying and analysing security protocols with automated tools. It adds a symbolic application of functions and equations. A process in calculus is used to describe concurrent computation, while in applied *pi* calculus, a process is a sequence of operations with a finite set T of functions and their arity, an infinite set of variables V, set of names N, and an equational theory Σ.

ProVerif [19] is an automatic proof verifying tool for analysing security protocols using applied *pi* calculus, supports user-defined equational theories and enables security properties verification. It also supports the theory of abstraction and uses applied *pi* calculus [28] as a language to formally describe and model security protocols. The syntax and informal semantics allow reasoning with protocols, supporting different cryptographic primitives, modelled by equations and rewrite rules. Additionally, it takes authentication, secrecy, and observational equivalence as security properties proved as input. This is translated into an internal representation of the protocol into crucial abstraction to an unbounded number of sessions [29]. Cryptographic primitives are modelled with functions, while messages are represented by terms relying on an infinite set of names $a, b, c, ...$, set of variables $x, y, z, ...$ and a finite set of function symbols $f1, \ldots, fn$. A set of reduction rules describe the semantics of each language construct in the form of function symbols and terms. The syntax and grammar of the ProVerif process are shown in Table 1 and more details can be found in [19]. This formal approach has been used to verify security properties assurance for 5G authentication [6,7,9,30], authorization [8] protocols.

Protocol modelling in ProVerif consists of declarations, macros, and main processes. To ensure that a protocol's correctness and secrecy are maintained, queries are used. With the ProVerif code, a protocol can be specified concisely using functions, queries, events, and types declaration. In association with free variables like free names that are known to the public and bound names that are only known by the process locally. If used as private, the names are excluded from the attacking vector [19].

2.2. Network architecture

The specification requires a communication system, so we adopt the 5G system model presented in [6,8] as shown in Fig. 1. They define the network entities as follows:

- User Equipment (UE): Is the end user accessing the service.
- Session Management Function (SMF): Is the home network SMF that communicates with the authentication server and outside network entities to establish a connection.
- Identity provider (IdP): Provides, manages federated identities and carries out federated authentication.
- SPAAA: Hosts the Authentication, Authorization, and Accounting (AAA) servers owned by third-party service providers.
- Service Server (SS). Hosts the protected services.

We demonstrate the protocol specification and modelling using two 5G protocols; SAP-AKA [9], and NS-FId [8] as shown in Tables 2 and 3, respectively. They have been formally verified using ProVerif and applied *pi* calculus, whose operatives are written as functions and processes following:

- Function:

```
fun f1(key, bitstring):bitstring.
fun nonce_to_bitstring(nonce):
bitstring.
fun aencrypt(bitstring, pkey):
bitstring.
fun hash(bitstring, bitstring):
bitstring
fun pk(skey): pkey.
```

Table 3
NS-FId protocol specification.

Msg1.UE→SMF:({ServName, SID}, {KAMF})
Msg2.SMF→ UE:({GPSI, SPID}, {KAMF})
Msg3.UE→ SPAAA:({ServName, SID, SPID},{PKSP})
Msg4.SPAAA→UE:({AuthzGrant, EID, KUE3A}, {PKUE})
Msg5.UE→IdP:({AuthzGrant, R1, GPSI},{PKIdP}
Msg6.IdP→UE:({FID, IDT, (hash(IDT), SKIDP), R1, (hash(FID, IDT, (hash(IDT), SKIDP), R1), SKIDP)}, PKUE)
Msg7.UE→ SPAAA:({IDT, (hash(IDT) SKIDP)}, {KUE3A})
Msg8.SPAAA→ UE:({AcT, ((hash(AcT); SKAAA), KUESS),(hash(AcT,(hash(AcT), SKAAA), KUESS), SKAAA)}, {KUE3A})
Msg9.UE→ SS:({AcT,(hash(AcT),SKAAA)},{KUESS})
Msg10.SS→UE:({SERV,(hash(SERV),SKSS},{KUESS})

Table 4
Enhanced operation semantics.

$M ::= terms \in \mathcal{M}$	
n	Name $(n \in \mathcal{N})$
x	Variable $(x \in \mathcal{X})$
$\{M_1, \ldots, M_k\}_{M_o}$	Encryption $(k \geq 0)$
$M ::= processes \in \mathcal{P}$	
0	Nil
$\langle M_1, \ldots, M_k \rangle.P$	Output
$(M_1, \ldots, M_j; x_{j+1}, \ldots x_k).P$	Input (with matching)
$P1 \mid P2$	Parallel composition
$(\nu n)\ P$	Restriction
$A\ (y_1, \ldots, y_n)$	Constant definition
$\{M_1, M_j; x_{j+1}, x_k\}_{M_o}$	Decryption

- Processes: The main process and process macros are used to encode the security entities and their sub-processes. The SAP-AKA protocol consists of processes `procUE` for the UE `procAAA` for the SPAAA, and `procSMF` for SMF. The NS-FId protocol consists of processes `procUE` for UE, `procSMF` for SMF, `procIDP` for IdP, `procSPAAA` for SPAAA, `procSS` for SS. These processes represent system entities with different parameters to run several sessions of the roles as state transitions.

Enhanced operation semantics and labels are used to associate rates to applied *pi* calculus processes as communication systems. The actual values are obtained with the provision of supplementary information on the architecture and supporting cryptographic schemes in relation to the system model. It interprets how algebraic processes and associates cryptographic operatives with quantitative measures. We use applied *pi* calculus for specifying our protocols, it is easier to estimate and evaluate their cost. The interpretation of the quantities is associated with transitions, including rates and measures as a cost of using cryptographic primitives. This method supports the user's input when analysing the activities and performance of the protocols, similar to a formal verification process.

3. Performance modelling

3.1. Semantic operations

We recall the applied *pi*-calculus grammar used in ProVerif [19], the formal specifications of the protocols use channel c as the communication channel, which is associated with input and decryption. The ProVerif syntax comprises terms M, N and processes P, Q, which denote sets of names with variables, respectively, and encryptions of tuples of terms $M1, ..., Mk$. With these main semantic, $N(M).Q$ is used to send message M on channel N, which is received by $N(x).P$. X is replaced with message M and the process Q stays in parallel with P after the communication. The enhanced operational semantics in Table 4 are used on top of ProVerif Syntax, consisting of processes, encryption and substitutes.

The process is prepared to perform reductions by defining semantics based on its configurations and a reduction relation [31], which is simply the definition of correspondences. A semantic configuration consists of a finite set of names E and a finite multiset of closed processes P. Through which all free names of processes in P must exist in environment E. The configuration $a_1, \ldots, a_n, P_1, \ldots, P_n$ corresponds intuitively to process $(\nu a_1) \ldots (\nu a_n)\ (P_1 \mid \ldots \mid P_n)$, creating a new name and executing P. The computation of evaluating a process in ProVerif is simplified by a reduction [19]. The meta-variables μ_{out} and μ_{in}, resp are used to denote runtime prefixes, whereby the enhanced operational semantics are built on the top of a reduction [20].

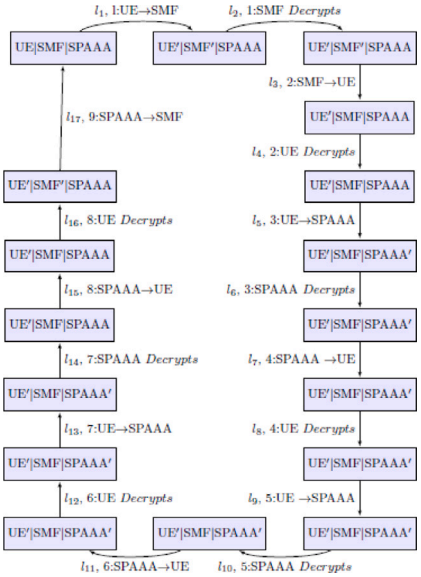


Fig. 2. SAP-AKA state transition system.

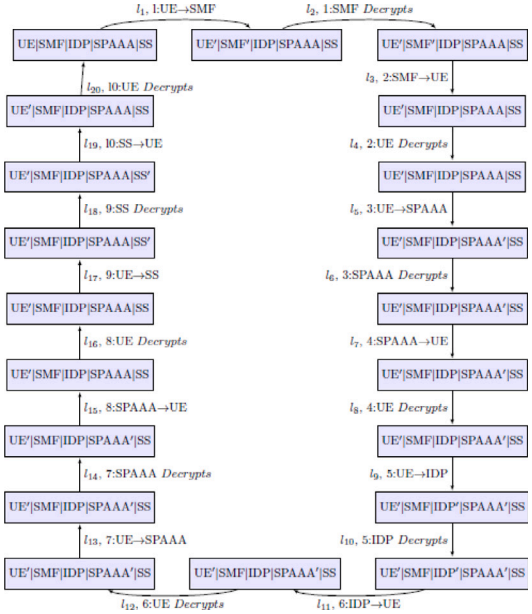


Fig. 3. NS-FId state transition system.

The process $\nu n.P$ makes a new private name n to act as P. It is abbreviated if $M = N$ then P when Q is 0. Now $N(x).P$ input from channel N, to run P with a formal parameter x replacing the actual message, with $\bar{\mathrm{N}}\langle M\rangle.P$ ready to output M on channel N and run P, if P is 0 it may be omitted. The focus is on normal semantics, new processes and labels that enhance transitions. Using constructors and destructors [32], the data structure can be represented as tuples and cryptographic operations for encryption and decryption, hence modelling perfect symmetric and asymmetric cryptography.

The parallel composition (|) of the processes makes up the whole system, as each process performs a specific number of activities before restarting. A restriction operator νnP is part of new names created, that goes about as a static binder for n in the process P. The communication between entities translated into a process can be shown by using the transition states of the system with enhanced labels and computing of the cryptographic primitives in the messages exchanged between entities as shown in Figs. 2 and 3. The entities are principals which are running in parallel. ν is a binder which binds the key with the principals. The processes are extended with active substitutions in Table 5, written $\{^{M}/_{x}\}$ where the variable x is replaced with the term M. Additionally, with

Table 5
Processes extensions with active substitutions.

$A, B, C ::=$	Extended processes
P	Plain processes
$A \mid B$	Parallel composition
$\nu n.A$	Name restriction
$\nu x.A$	Variable restriction
$\{^{M}/_{x}\}$	Active substitution

$\{^{M}/_{x}\}$ taken as a process, equivalent to let $x = Min\ldots$, which is useful by adding a restriction: $\nu x.(\{^{M}/_{x}\}|P)$ corresponding to let $x = M$ in P [21]. The principals receive and send messages that include terms, associated with input that are directly embodied in $M_1, ..., M_k$ and $(M_1, ..., M_j; x_{j+1}, ..., x_k)$.

3.2. Enhanced labels

After specification, an enhanced label is associated with each transition i.e., each communication and decryption [20], by using enhanced operational semantics based on labelled bisimilarity [21]. The enhanced label records communication output and components of the input with syntactic contexts, facilitated by processes generating finite state spaces. With labelled operational semantics, an enhanced label is associated with each transition of the ProVerif process, e.g.,`((!procUE()) (!procSMF()) (!procSPAAA()).` A context label ϑ is associated with each prefix of a given process. Context labels on a given process are distributed by $\triangleright$ and then $\mathcal{T}$ constructs the labels with a $\| 0$ (resp.$\| 1$) for the left branch of parallel composition and each restriction of the name a is built with a νa. This leads to new processes LP, including T, T', with each prefix associated by a context label as defined in [20]. The mapping of processes from P in LP is defined by T.

Definition 1.
Set Θ of enhanced labels, includes θ and defined by Eq. (1)

$$\theta ::= \underbrace{\langle \vartheta \|_{1-i} \vartheta_O \mu_O, \vartheta \|_i \vartheta_I \mu_I \rangle}_{out \quad in} | \underbrace{\langle \{M'_1, \ldots, M'_j; x_{j+1}, \ldots, x_k\}_{M'_0} \rangle}_{dec} \tag{1}$$

Labelled transitions $\xrightarrow{\vartheta}$ occurs when an output T is the same as an input T'. That is, matching the term from encryption M_o against the decryption pattern $M\prime_o$. The reduction $\xrightarrow{\vartheta}$ ends under parallel composition and restriction and is only used for communication labels. With ProVerif processes, the output or input prefixes are improved with sequences ϑ of tags such as ν_n, $\|_0$ or $\|_1$. If the prefix appears after a restriction, the tag ν_n will occur in a sequence and tag $\|_0$ (resp$\|_1$) occurs, if the prefix is moved to the left of a parallel composition [20].

Example 1. The state transitions of the protocol are predated by the sequence of tags which can be reduced as:

- $\vartheta_{UE} = \nu_{PKUE}\nu_{PKSP}\nu_{ServName}\nu_{SID} \| 0 \| 0$ predating the prefixes of UE.
- $\vartheta_{SMF} = \nu_{PKUE}\nu_{PKSP} \| 0 \| 1$ (resp. $\vartheta'_{SPAAA} = \nu_{PKUE}\nu_{PKSP}\nu_{GPSI}\nu_{SPID} \| 0 \| 1$ predating the first input of SMF.
- $\vartheta_{SPAAA} = \nu_{PKUE}\nu_{PKSP} \| 1$ (resp. $\vartheta' SPAAA = \nu_{PKUE}\nu_{PKSP}\nu_{Identity}\nu_{KUESP} \| 1$) preceding the first input of $SPAAA$.

In the transition system graph, the processes are the nodes and arcs are possible transitions between the nodes. Labelled operational semantics enable reasoning about processes, states and transitions. The labelled semantics define a relations $P \xrightarrow{(\alpha)} P'$ referring to transitions between state P and P', represented as $P \xrightarrow{(label,caption)} P'$, the multi states transitions are presented as $P \xrightarrow{(label,caption)} P'^1, P'^2, P'^n$.

- Label $p(M)$, M refers is a term with names and variables to correspond to an input of M on a.
- Label $\bar{a}\langle u\rangle$ or $\nu u.\bar{a}\langle u\rangle$, with a channel name or a variable type u corresponds to an output of u on a.

This represents the label of the transition and the part of the protocol in transition, such as 1 : UE→SMF, for the communication between UE and SMF.

3.3. Defining the transition cost

The cost function $(.) is used to assign a cost to a transition derived from label [20]. The cost is any quantitative measure that can impact the properties of transitions such as cryptographic procedures that perform encryption and decryption on these systems. The cost of transitions is derived by inspecting enhanced labels and measured by considering the time that the system might take to stay within a specific transition. The time overhead is used by the function to specify the cost of a protocol's primitives action.

Each transition labelled by θ gets a cost associated with it by the cost function, representing the rate of the transition. This parameter representation indicates the exponential distribution of time measurement of θ [20]. Hence, action μ and context ϑ determine the cost of the elements of an enhanced label $\vartheta\mu$. Moreover, a scaling factor r is introduced in correlation with each procedure of the transition θ under consideration. The costs are assigned to terms and components of activities μ_{in}, μ_{out} represented by functions as follows [20].

Table 6
Cost description.

Term	Description
n	Size of the message
m_i	Size of the ith encryption
e	Cost of unitary encryption
d	Cost of unitary decryption
s	Cost of unitary output
l_i	Label in relation to the state
c_i	Cost in relation to the label

- Unary terms cost is defined by $f_u(n)$.
- The cost function that calculates the costs of the procedures for encryption methods is defined by f_{enc}.
- The procedure cost for sending and receiving cryptographic primitives is defined by f_{in} and f_{out}.
- The cost function for matching patterns of size j is defined by $f = (j)$.
- The cost of encryption or decryption methods defined by $f_{kind}(crypt)$.
- The cost of a specific key is defined$f_{kind}(M_O)$.
- The cost of the size of the cleartext to encrypt is defined by $f_{size}(ctxt)$.
- The cost of the sent or received message in relation to the cost evaluation of an output/input is defined $f_{size}(msg)$ [20]. The computation cost using $f = (j)$ for j terms is.

$\$_T(n) = f_u(n)$
$\$_T(x) = f_u(x)$
$\$_T(\{M_1, \ldots, M_n\}_{M_0}) = f_{enc}(f_{kind}(crypt), f_{size}(ctxt),$
$f_{kind}(M_0), \$_T(M_1, \ldots, M_k))$
$\$_T((M_1, \ldots, M_n) = min\$_T(M_1), \ldots, \$_T(M_n))$
$\$_{in}(\mu_{in}) = f_{in}(f_{size}(msg), f = (j), \$_T(M_1, \ldots, M_j))$
$\$_{out}(\mu_{out}) = f_{out}(f_{size}(msg), \$_T(M_1, \ldots, M_k))$

The evaluation of parallel composition is based on the number np of processes available. With $\$_o(\|) = 1$ as a specific case for an unbound number of ProVerif processes. The number of names $n(P)$ of the process P determines the cost restriction. It also depends on the name $f_{kind}(a)$ such as nonce, key, hash function, and MAC. Therefore, the label of the transition is $\langle \vartheta \|_i \vartheta_{in}\mu_{in}, \vartheta \|_{1-i} \vartheta_{out}\mu_{out}\rangle$ and these are recorded in ϑ_{in} and ϑ_{out}, constructed by applying function T. This results in $\langle \|_i \vartheta_{in}\mu_{in}, \|_{1-i} \vartheta_{out}\mu_{out}\rangle$ pairing, which corresponds to the real communication. In addition, an exponential distribution with rate r induces the time parameter Δt that is required to have a probability closer to 1 in relation to the Markov Chain process. By The estimation of the correspondence duration with a fixed rate $r = min\{\$_{in}(\|_i \vartheta_{in}\mu_{in}), \$_{out}(\|_{1-i} \vartheta_{out}\mu_{out})\}$ as a minimum cost, the communication occurs at the same time. While the operations are registered in ϑ, which accounts for the operations' common context. The cost is computed with f_{dec} to derive the decryption algorithm cost if the label is $\langle dec\rangle$. Therefore, induction on θ defines the cost by using the functions $\$\mu$ as basis, and then $\$_o$.

Definition 2. This is where the cost function is i = 0, 1 as defined by [20] $\$(\mu) = \$\mu(\mu)$
$\$(o\theta) = \$o(O) \times \$(\theta)$
$\$(\|_i\ \theta) = \$o(\|\ i) \times \$(\theta)$
$\$(\langle \vartheta \|_i \vartheta_{in}\mu_{in}, \vartheta \|\ 1 - i\vartheta_{out}\mu_{out}\rangle) = \$(\vartheta) \times \min\$(\|_i \vartheta_{in}\mu_{in}), \$(\|_{1-i} \vartheta_{out}\mu_{out})$
$\$(\langle dec\rangle) = f_{dec}(f_{kind}(crypt), f_{size}(ctxt), f = (j), f_{kind}(MO), \$T(M_1, \ldots, M_j))$

Next is the fine-tuning of probabilistic distribution concerning the anticipated speed of actions by performing operations on costs. The cost is influenced by the following factors:

- The input, output components and context determine the cost of communication. While the size of the message and the cost of message segments determine the cost of output.
- The algorithm, size of the cleartext, and type of the key, collectively determine the encryption cost.
- The size of the message with the cost of terms and the number of checks made for message acceptance determine the cost of input.
- The algorithm, ciphertext size and type of the key determine the ciphertext decryption cost.
- The cost of decryption is not determined by its context, it all depends on the number of checks made to accept the decryption.

A set of parameters of the cost function are used to reflect on the architecture and encryption scheme, taking into account the number of processes and cryptographic algorithms [20]. The cost is only taken into account because of the parallel composition. Moreover, the number of processors available np is used to analyse parallel composition. The cost of restriction is determined by the process P's number of names $n(P)$. It also depends on the name, such as nonce, long-term key.

Now that every transition in the transition systems comes with a cost, we ignore the cost for simplicity due to limitations in computation performance. We can also assign cost 1 to each tag $\|\ i$ (i = 0, 1) because we can assume that each principle has its

processing unit. The context is ignored, and the same cost is given to output and input. A transition communication is assigned a cost equal to $n * s + \sum_{i=1}^{l} m_i * e$, decryption is assigned a cost equal to $n * d$ and terms are described in Table 6.

Example 2. The cost of the third transition of SAP-AKA protocol, Fig. 2, with label

$l_3 = \langle \vartheta_{UE}(\{ServName, SID, SPID\}_{PK_{SP}}), \vartheta_{SPAAA}(t_{enc}^{UE})\rangle$ (msg3).

The cost is 3s + 3e = C, The output message has a cost of 3s, and the encrypted cleartext has a cost of 4e. The cost of decryption of this message in the third transition is 3d, that is, decrypting a ciphertext back to the cleartext of 3e. The complete list of the protocols' costs c_i in relation to their labels l_i is presented in Section 5. The cost parameters vary due to the different system architectures, protocols, encryption schemes, algorithms and cryptographic primitives used. In 5G, for instance, the Elliptic Curve Cryptography (ECC), Sequence (SQN), Authentication and Key Agreement AKA challenge, XOR must be considered. The cost of communication and encryption is affected by the speed of operations and the communications link.

4. Continuous time Markov chain

This section explains how to get the quantitative data needed to derive a continuous time Markov Chain (CTMC) [18] using enhanced operational semantics. The CTMC compromises a set of states and labelled transitions between the states and a succession of random values whose probabilities at a given time interval are dependent on the previous states' values [33]. As explained earlier, a function is used to assign costs to individual transitions, enabling applications to tailor a probabilistic distribution based on costs. The costs are interpreted as parameters of exponential distributions [20]. Because the arcs that share source and target are collapsed when the exponential distributions of transitions are determined, it leads to a numerical process. Therefore, cost represents the rate r of the transition, that is the exponential distribution of the duration times of the transition.

The next transition appearance does not depend on when the last transition appeared. Because all transitions are believed to be homogeneous, the rate of transition is unaffected by the passage of time. As a result, the parameter r is linked to a transition in order to determine certain transition probabilities or the rate at which a system switches from acting like process P_i to acting like process P_j. Hence, its equivalence to the sum of all the costs of all feasible transitions from P_i to P_j. Furthermore, because each pair of nodes has only one transition, rates inside a transition system match with single costs.

Definition 3. The rate at which transitions between two states P_i and P_j occur is expressed as $q(P_i, P_j)$ as shown in Eq. (2) as defined in [20].

$$q(P_i, P_j) = \sum_{P_i \xrightarrow{\theta_k} P_j} \$(\theta_k). \tag{2}$$

A directed graph is used to depict a CTMC C, with the nodes representing the states of C and the arcs connecting only the states that are reachable from one other. The rates at which the process leaps from one state to the next can be arranged in a square matrix Q, referred known as the generator matrix. It is the adjacency matrix of the graph representation of the process's CTMC ($CTMC(P)$). The instantaneous transition rates denoted in Eq. (3) are the entries of Q [20].

$$q_{ij} = \begin{cases} q(P_i, P_j) = \sum_{P_i \xrightarrow{\theta_k} P_j} \$(\theta_k) & \text{if } i \neq j \\ -\sum_{\substack{j=1 \\ j \neq i}}^{n} q_{ij} & \text{if } i = j \end{cases} \tag{3}$$

After long periods of execution, the performance measures of systems become comprehensible. Because they have finite and cyclic features, these measures of process P are derived by leveraging the stationary probability distribution Π for the CTMC and coupling it with P [20].

Definition 4. If $\Pi^t(x_i) = p(X(t) = x_i)$ is the probability that a CTMC is in the state x_i at time t, and allow $\Pi^0 = (\Pi^0(x_0), \ldots, \Pi^0(x_n))$ becomes the initial distribution of states $x_0, x_1, \ldots, x_n$. Therefore, the stationary probability distribution of CTMC is $\Pi = (\Pi(x_0), \ldots, \Pi(x_n))$ is shown in Eq. (4)

$$\Pi Q = 0 \; and \; \sum_{i=0}^{n} \Pi(x_i) = 1. \tag{4}$$

The solutions of the system linear equations are the stationary distribution for each of the systems, according to Definition 4. To exploit the preferred numerical package available for needed computations and stochastic analysis, standard numerical techniques are used. A reward structure is used to assess the performance of a process P [33]. The performance model's reward structure is a function that associates a value with any state that is passed through in a calculation of P [22].

Definition 5. Given a function $\rho\theta$ that corresponds to each transition θ in a transition system as a transition reward [20], the reward of a state P is defined in Eq. (5)

$$\rho p = \sum_{P \xrightarrow{\theta} Q} \rho\theta. \tag{5}$$

A process's reward structure P is a reward vector with as many elements as the number of derivatives of the process P. From it, as well as the stationary distribution Π of the $CTMC$ of a process, performance measures P were calculated.

Definition 6. Let the stationary distribution of $CTMC(P)$ be Π. The reward for P with Π is calculated in Eq. (6) as defined in [20].

$$R(P) = \sum_{P_i \in d(P)} \rho P_i \; X \; \Pi(P_i). \tag{6}$$

The total of the values of Pi multiplied by the matching reward structure is obtained when an encryption technique is used. This adds up to, taking into account the time spent in states where the encryption technology is enabled.

We can calculate rewards using rates of transitions, even though the reward structure is only a function that correlates a reward with a state moving through computation of process P [22]. By measuring the system's throughput in terms of the amount of work accomplished per unit of time using non-zero reward value against the rate of corresponding transition [34].

Definition 7. Let process P reward structure be $\rho\theta = \rho\theta(0), \ldots., \rho\theta(n-1)$. Process P's total reward is computed as Eq. (7).

$$R(P) = \sum\nolimits_i \quad \rho(i).X_i. \tag{7}$$

$$\mathbf{Q1} = \begin{array}{c} \\ l1 \\ l2 \\ l3 \\ l4 \\ l5 \\ l6 \\ l7 \\ l8 \\ l9 \\ l10 \\ l11 \\ l12 \\ l13 \\ l14 \\ l15 \\ l16 \\ l17 \end{array} \begin{array}{c} \begin{array}{cccccccccccccccc} l1 & l2 & l3 & l4 & l5 & l6 & l7 & l8 & l9 & l10 & l11 & l12 & l13 & l14 & l15 & l16 & l17 \end{array} \\ \begin{pmatrix} -b & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2d & 2d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3d & 3d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2d & 2d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -d & d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a & a & 0. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -d & d & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5d & 5d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -b & b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2d & 2d & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3d & 3d \\ s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -s \end{pmatrix} \end{array}$$

$$\Pi_1 = \left[\frac{A}{b}, \frac{A}{2d}, \frac{A}{c}, \frac{A}{3d}, \frac{A}{b}, \frac{A}{2d}, \frac{A}{a}, \frac{A}{d}, \frac{A}{a}, \frac{A}{d}, \frac{A}{g}, \frac{A}{5d}, \frac{A}{b}, \frac{A}{2d}, \frac{A}{c}, \frac{A}{3d}, \frac{A}{s}\right]$$

5. Performance evaluation based on analytical model approach

Having specified the protocols, and defined the labelled enhanced operation semantic and cost function, this section evaluates the performance of 5G protocols using the Markov Chain process and other mathematical techniques. We use SAP-AKA [9] and NS-FId [8] as our case studies. For the protocols, the stationary distributions of the Markov Chain $\Pi_i = (X_0, \ldots, X_{n-1})(i = 1, 2$ and $n = 6, 8)$ are used as solutions, with the following linear equation for each protocol [20] as illustrated in Eq. (8).

$$\Pi Q = 0 \; and \sum_{i=0}^{n-1} X_i = 1. \tag{8}$$

5.1. SAP-AKA

The state of transition and labels are shown in Fig. 2, and the cost and transition association in Tables 7 and 8. Consider the transition system which is both finite and cyclic at the beginning of a state to guarantee that it has stationary distributions and the following generator matrix $Q1$ = CTMC (SAP-AKA) is derived, and the stationary distribution is Π_1, where $A = 20s + 19e + 19d$.

Table 7
Cost labels for the protocols.

SAP-AKA	NS-FId	5G-AKA
c1 = 2s + 2e	c1 = 2s + 2e	c1 = 2s + e
c2 = 2d	c2 = 2d	c2 = 3s
c3 = 3s + 3e	c3 = 2s + 2e	c3 = 3s
c4 = 3d	c4 = 2d	c4 = d
c5 = 2s + 2e	c5 = 3s + 3e	c5 = 5s + 7e
c6 = 2d	c6 = 3d	c6 = 5s
c7 = s + e	c7 = 3s + 3e	c7 = 5s
c8 = d	c8 = 3d	c8 = 4d
c9 = s + e	c9 = 3s + 3e	c9 = s+e
c10 = d	c10 = 3d	c10 = s
c11 = 5s + 5e	c11 = 5s + 5e	c11 = 2s
c12 = 5d	c12 = 5d	c12 = s
c13 = 2s + 2e	c13 = 2s + 2e	c13 = d
c14 = 2d	c14 = 2d	
c15 = 3s + 3e	c15 = 4s + 4e	
c16 = 3d	c16 = 4d	
c17 = s	c17 = 2s + 2e	
	c18 = 2d	
	c19 = 2s + 2e	
	c20 = 2d	

Table 8
Metrics variables.

Variable	Description
a	s + e
b	2s + 2e
c	3s + 3e
f	4s + 4e
g	5s + 5e
h	6s + 6e
i	7s + 7e

5.2. NS-FId

The state of transition and labels is shown in Fig. 3, and the cost and transition association in Tables 7 and 8. Consider the transition system, the following generator matrix $Q2$ = CTMC (NS-FId) is derived, and the stationary distribution is Π_2, where $B = 28s + 28e + 28d$.

$$\mathbf{Q2} = \begin{array}{c} \\ l1 \\ l2 \\ l3 \\ l4 \\ l5 \\ l6 \\ l7 \\ l8 \\ l9 \\ l10 \\ l11 \\ l12 \\ l13 \\ l14 \\ l15 \\ l16 \\ l17 \\ l18 \\ l19 \\ l20 \end{array} \begin{array}{c} \begin{array}{cccccccccccccccccccc} l1 & l2 & l3 & l4 & l5 & l6 & l7 & l8 & l9 & l10 & l11 & l12 & l13 & l14 & l15 & l16 & l17 & l18 & l19 & l20 \end{array} \\ \begin{pmatrix} -b & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2d & 2d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -b & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2d & 2d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -c & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -3d & 3d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -c & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3d & 3d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c & c & 0. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3d & 3d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5d & 5d & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -b & b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2d & 2d & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -f & f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4d & 4d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -b & b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2d & 2d & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -b & b \\ 2d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2d \end{pmatrix} \end{array}$$

Table 9
Protocols performance evaluation.

Protocols	Efficiency	Throughput
3GPP-5G-AKA	$\frac{3GPPP-5G-AKA}{3d}$	$\frac{3d}{27s+9e+9d}$
SAP-AKA	$\frac{A}{8d}$	$\frac{3d}{20s+19e+19d}$
NS-FId	$\frac{B}{10d}$	$\frac{2d}{28s+28e+28d}$

$$\Pi_2 = \left[\frac{B}{b}, \frac{B}{2d}, \frac{B}{b}, \frac{B}{2d}, \frac{B}{c}, \frac{B}{3d}, \frac{B}{c}, \frac{B}{3d}, \frac{B}{c}, \frac{B}{3d}, \frac{B}{g}, \frac{B}{5d}, \frac{B}{b}, \frac{B}{2d}, \frac{B}{f}, \frac{B}{4d}, \frac{B}{b}, \frac{B}{2d}, \frac{B}{b}, \frac{B}{2d}\right]$$

5.3. Analytical results

This section illustrates the cost of the protocols' communication based on the systems labels and values of exchanged packets of data (protocol messages) between devices (protocol entities).

5.3.1. Efficiency

We present the protocol's relative efficiency in terms of cryptographic procedure use. Any transition in which decryption is enabled receives a value of 1, a non-zero transition reward, while any other transition receives a value of 0. We give the following items a 1:

1. the 2nd, 4th, 6th, 8th, 10th, 12th, 14th, 16th transitions in SAP-AKA
2. the 2nd, 4th, 6th, 8th, 10th, 12th, 14th, 16th, 18th, 20th transitions in NS-FId

Using the performance measure R, the performance of the protocols is below:

$$R(SAP-AKA) = \frac{A}{8d} \tag{9}$$

$$R(NS-FId) = \frac{B}{10d} \tag{10}$$

It is possible to prove that one protocol is more costly than the other, for each positive s, d and e, depending on the encryption scheme and having used the same quantitative measure for performance evaluation. We can also measure and evaluate the efficiency of multiple models of the same protocol.

5.3.2. Throughput

The throughput is the result of associating a transition reward to a rate and a transition of an activity. Since a transition is run once in a system, the CTMC is cyclic and a label corresponds to the different transactions, the throughput of all transactions is the same. We choose the last transaction to compute the throughput of the protocol/system by associating a transition reward equal to the rate with the previous protocol communication and then giving zero transition reward to all the other communications. Assuming that encryption and decryption have the same cost, point multiplication takes longer than decryption. As mentioned in [35], a cryptographic scheme's energy consumption is linked to its temporal complexity, as the results indicate.

The reward structure and total rewards are computed as follows:

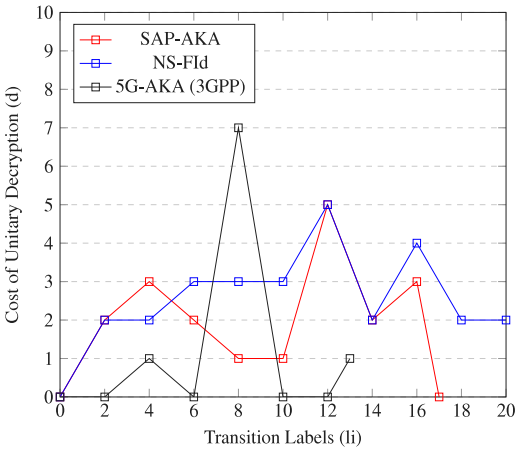
$$\begin{aligned} \rho 1 &= (0.....C_{16}), \quad (C_{16}) = 3d, \\ R(SAP-AKA) &= \frac{3d}{20s+19e+19d} \end{aligned} \tag{11}$$

$$\begin{aligned} \rho 2 &= (0.....C_{20}), \quad (C_{20}) = 2d, \\ R(NS-FId) &= \frac{2d}{28s+28e+28d} \end{aligned} \tag{12}$$

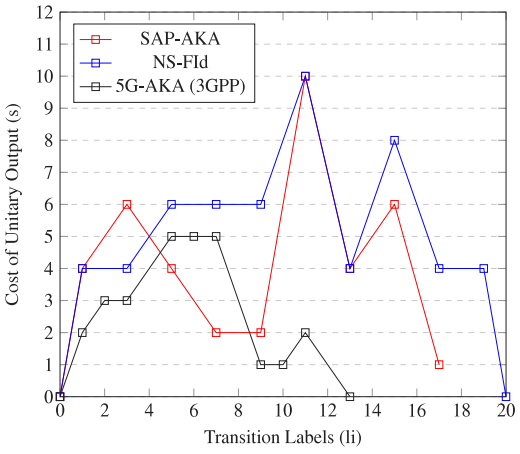
5.4. Analysis

As mentioned earlier, the 5G-AKA protocol is used as a benchmark to compare the performance results of SAP-AKA and NS-FId protocols, shown in Table 9. It was defined in [36] and formally analysed in [6]. The stationary distribution is $3GPP-5G-AKA = 20s+13e+13d$. The results in Table 9 and Fig. 4 indicate that 5G-AKA is more efficient and with better throughput than the SAP-AKA and NS-FId, this is due to additional protection as 5G-AKA cannot be used outside the HN such as non-repudiation and single sign-on (SSO), while these protocols can.

Any activity was supposed to be exponentially distributed, but generic distributions are also possible [37], since they rely solely on enhanced labels. It is easier to calculate the CTMC associated with a process's transition system once rates have been assigned to transitions. We used a continuous time approach to evaluate the process' performance based on its stationary distribution, if



(a) Protocols Effeciency



(b) Protocols Throughput

Fig. 4. Efficiency and throughput comparison of 5G protocols based on analytical modelling.

any, and adopted the quantification of transitions approach presented in [20] but for security properties and bisimilarity labelling, we used ProVerif and applied *pi* calculus respectively. The evaluation is based on operational semantics and labels for security behavioural and quantitative analysis. Numerical results illustrate the effectiveness of the model used, system linear equations are used to calculate some results and to make analytical modelling possible with some assumptions.

6. Performance evaluation based on a simulation model approach

This section also evaluates the performance of SAP-AKA and NS-FId protocols as presented in [8,9], respectively, using a simulation model. We intend to measure the network impact of these protocols and compare them with the 5G-AKA protocol as presented in [6]. Additionally, simulation and analytical model results are compared and analysed to check the effectiveness of the proposed approach.

6.1. Simulation environment settings

In order to perform protocol simulation, the NS-3 tool is installed and configured on an Ubuntu Linux virtual machine in the VirtualBox environment installed on a Windows computer. The implementation and testing environment are as follows:

- Windows 10: Processor Intel i7 - 2.40 GHz, 16 GB RAM and 250 GB disk space
- VirtualBox
- Ubuntu 64 bit operating system
- NS-3.33

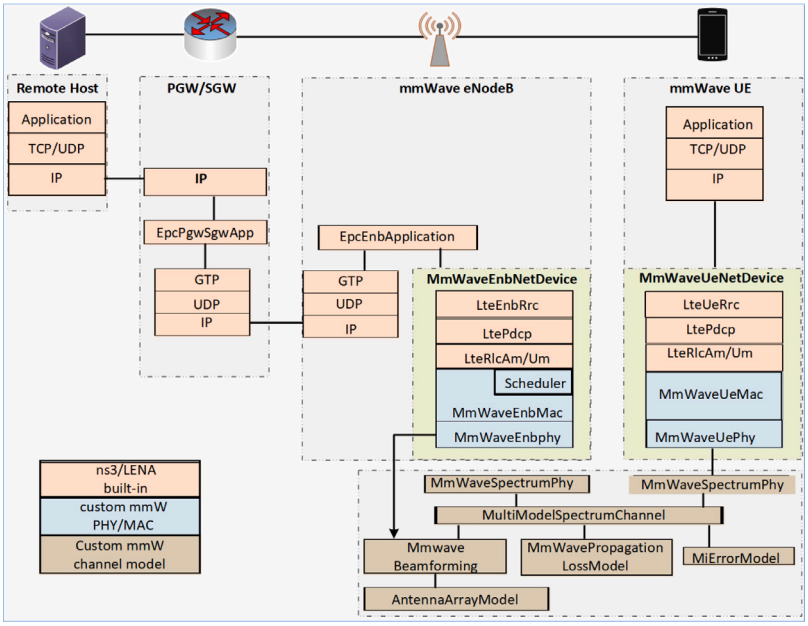


Fig. 5. NS-3 simulation structure composed of 5G mmWave and LTE models.

```
ededris@ededris-VirtualBox:~/repos/ns-3-allinone/ns3-mmwave$ ./waf --run 5gaka
Waf: Entering directory `/home/ededris/repos/ns-3-allinone/ns3-mmwave/build'
Waf: Leaving directory `/home/ededris/repos/ns-3-allinone/ns3-mmwave/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.060s)
UE=1, SEAF=1, AUSF=1, ARPF=1
Setup Complete.
Total packets received (UE=1, SEAF=1, AUSF=1, ARPF=1) : 4
```

Fig. 6. 5G-AKA NS-3 simulation.

```
ededris@ededris-VirtualBox:~/repos/ns-3-allinone/ns3-mmwave$ ./waf --run nsfid
Waf: Entering directory `/home/ededris/repos/ns-3-allinone/ns3-mmwave/build'
Waf: Leaving directory `/home/ededris/repos/ns-3-allinone/ns3-mmwave/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.280s)
UE=1, SMF=1, IDP=3, SPAAA=3, SS=3
Setup Compete.
Total packets received (UE=1, SMF=1, IDP=3, SPAAA=3,SS=3) : 10
```

Fig. 7. NS-FId NS-3 simulation.

```
ededris@ededris-VirtualBox:~/repos/ns-3-allinone/ns3-mmwave$ ./waf --run sapaka
Waf: Entering directory `/home/ededris/repos/ns-3-allinone/ns3-mmwave/build'
Waf: Leaving directory `/home/ededris/repos/ns-3-allinone/ns3-mmwave/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.340s)
UE=1, SMF=1, SPAAA=4
Setup Compete.
Total packets received (UE=1, SMF=1, SPAAA=4) : 6
```

Fig. 8. SAP-AKA NS-3 simulation.

The simulation model was built using C++ programming language based on NS-3 5G mmWave module [15,38] to represent the current non-standalone implementation, 5G is being deployed with 5G radio technology and LTE core network. The NS-3 is made up of modules that are used to programme and run a successful simulation, which includes node, application, net device, and topology helpers [15]. In order to simulate 5G network communication, the nodes, net device, applications and topology helpers were modified to represent communication of 5G-AKA, SAP-AKA and NS-FId protocols entities based on the architecture shown in Fig. 1 as [6,8,9]. The NS-3 end-to-end simulation structure with mmWave eNB and UE radio stacks is shown in Fig. 5 [38]. To quantify the security properties, cryptographic operatives are defined in a message using the `ApplicationContainer`: `serverAppContainer` and `clientAppContainer` with `sendMessage ()` function for `UdpClient` to send a single message for every message included in authentication and authorization procedures [39]. 5G-AKA, NS-FId and SAP-AKA NS-3 simulation

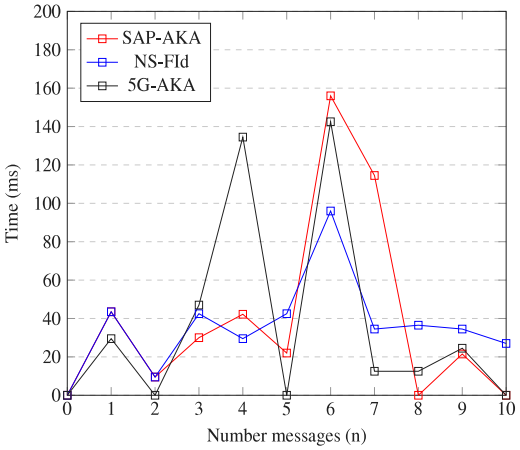


Fig. 9. Comparison of computational cost for 5G protocols based on simulation modelling.

Table 10
Approximate time for cryptographic operations.

Notation	Description (time to compute)	Computation time (ms)
T_{Av}	Authentication vector	33.5
T_h	Hash function	5
T_{Se}	Symmetric encryption	4
T_{Sd}	Symmetric decryption	5.5
T_{Ae}	Asymmetric encryption	8
T_{Ad}	Asymmetric decryption	9.5
T_{Tn}	Token	5
T_{Ts}	Timestamp	5
T_{KDF}	Key generation	12.0
T_E	Execute	21.5
T_V	Verify	12.5

Table 11
Cryptographic primitive size.

Primitive	Value
Symmetric key	128 bits
Asymmetric key	256 bits
SHA256	256 bits
Token	128 bits
Nonce	128 bits
5G IDs	64 bits
Nonce key	256 bits
D1	256 bits
Strings	32 bits
MAC	64 bits
SQN	48 bits
Timestamp	16 bits
RES	256 bits

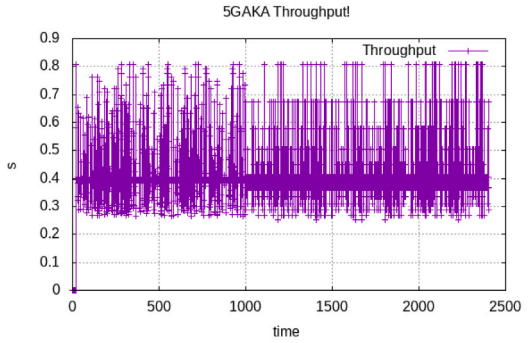
output is shown in Figs. 6, 7, 8, respectively. To run the simulation, `./waf --run scratch/sapaka` command is used on the terminal and `numUe` is defined as UE pointing to `numEnb` and `remostHost` installed on the node to run the simulation successfully.

6.2. Computational and communication cost

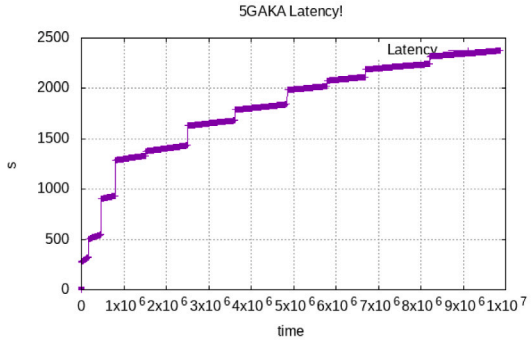
It has to be noted that the computational and communication costs of the protocols are evaluated with the assumption that all protocols are using 5G cryptographic primitives and algorithms recommended by 3GPP [6,36]. The evaluation of computational and communication costs of the 5G security protocols follows a similar method as presented in Section 5, the protocols are compared with the 5G-AKA protocol since it is the recommended network access security protocol for the 5G core network.

Table 12
Evaluation metrics.

Parameters	Values
Throughput	bits/ms
Latency	ms
m	Messages primitive cost
n	Total sum of m



(a) Throughput



(b) Latency

Fig. 10. Communication cost for 5G-AKA protocol.

6.2.1. Computational cost

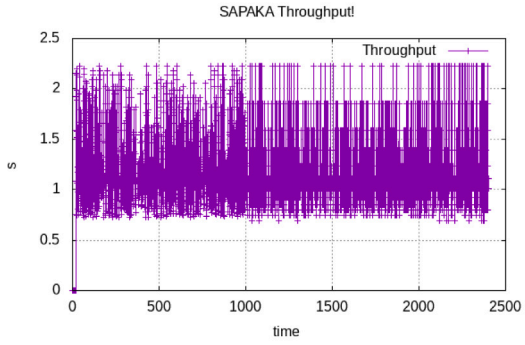
To evaluate the computation cost of 5G security protocol, the time cost of security vectors and primitives generation are defined as T_{AV}, T_h, T_{SE}, T_{SD}, T_{AE}, T_{AD}, T_{Tn}, T_{Ts}, and T_{KDF} as shown in Table 10. Moreover, these are the estimated times in milliseconds (ms) needed for computing the respective cryptographic primitives and messages. The estimate is based on ECC algorithm and size of the primitives, this is the time it takes for key generation, encryption, decryption, computing of the hash, timestamp, token and verification process. Fig. 9 shows the variation in computation times of the protocols.

6.2.2. Communication cost

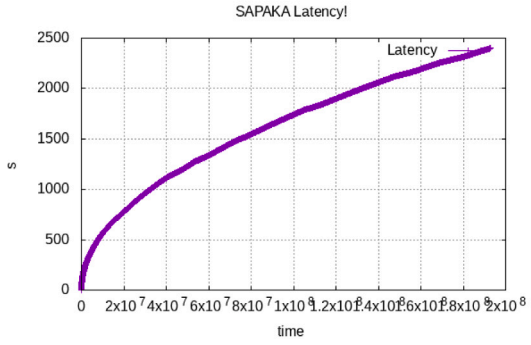
To evaluate the communication cost, the protocol cryptographic primitives, scheme and messages are used as parameters with values as shown in Tables 11 and 12, which are used to define the cost of a protocol. Security context used such as AMF, $synch\_fail/mac\_fail$ $authzgrant$ code, $dataname$, and $success$ message in 5G-AKA, NS-FId and SAP-AKA, respectively are represents as strings. The message sent between entities is defined as m and the n is the total sum of m in a protocol, $n = (m1, m2, m3, m4....)$ measured in bits. However, the value of n may vary depending on the number of messages and the primitives used. n is used to get the communication cost of the protocols by measuring the throughput (bits/ms) and latency (ms) as performance metrics during the protocol simulation in NS-3. Figs. 10–12 show the NS-3 output of protocols throughput and latency.

6.3. Simulation results

The protocols were simulated in NS-3.33 using a modified 5G mmWave module, based on the analysis of the trace pcap, and XML files generated in NS-3, the quantified output of the security protocols were obtained as a result. The values and metrics in



(a) Throughput



(b) Latency

Fig. 11. Communication cost for SAP-AKA protocol.

Table 13
Computational cost of the security protocols.

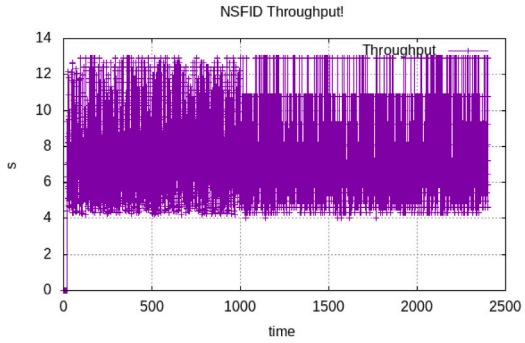
Protocols	Computational time (ms)	Total time (ms)
5G protocols		
SAP-AKA	$T_E + 6T_{Se} + T_{Ae} + 6T_{Sd} + T_{Ad} + 1T_{Av} + 13T_{KDF} + 6T_V$	439.2
NS-FId	$T_E + 6T_{Se} + 4T_{Ae} + 6T_{Sd} + 4T_{Ad} + 1T_{Av} + 2T_{KDF} + 5T_h + 2T_n + 10T_V$	396
5G-AKA	$T_E + 6T_{Se} + T_{Ae} + 6T_{Sd} + T_{Ad} + 1T_{Av} + 8T_{KDF} + 2T_h + 11T_V$	545.5

Tables 11 and 12 are used as inputs for NS-3 simulation to show results for SAP-AKA, NS-FId and 5G-AKA protocols, measuring throughput and latency. The simulation results also illustrate the exchange of packets of data (protocol messages) between the nodes (protocol entities) and the relevant packets received by the devices.

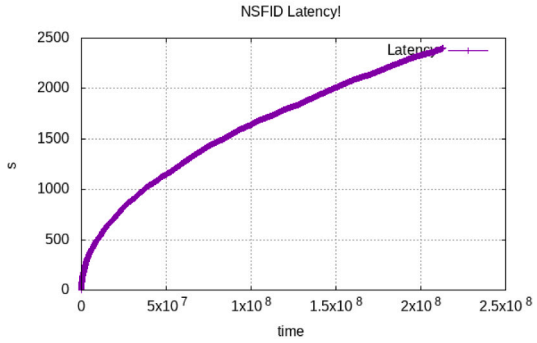
6.4. Analysis

Similarly, for simulation analysis, the 3GPP's 5G-AKA protocol [36] is used as a benchmark for evaluating 5G security protocols. We also have to consider the number of message exchanges between the entities during the protocol run, SAP-AKA has 9, NS-FId and 5G-AKA have 10 messages. The total computational cost of each security protocol is summarized in Table 13, with Fig. 9 illustrating the performance comparison between the security protocols. The SAP-AKA uses a similar computation time as NS-FId but between messages 6 and 7, it takes longer to compute as that is when the authentication vectors are generated and verified. For 5G-AKA, this occurs at message 4 and message 6. Generally, the results indicate that the SAP-AKA and NS-FId protocols have relatively similar computation costs as the 5G-AKA protocol.

The total communication cost of each security protocol is summarized in Table 14, with the plot graphs for throughput and latency generated directly from NS-3 simulation as shown in Figs. 10 and 11. With the total of messages exchanged between entities, the total communication cost is also relatively similar. Even though, the results indicate that SAP-AKA has lower communication cost than both NS-FId and 5G-AKA protocols. It is interesting to note that both NS-FId and 5G-AKA protocols have 10 m but have



(a) Throughput



(b) Latency

Fig. 12. Communication cost for NS-FId protocol.

Table 14
Communicational cost of the security protocols based on simulation modelling.

Protocols	Total communication cost (bits) (n)	Number of messages (m)
5G protocols		
SAP-AKA	3136	9
NS-FId	5472	10
3GPP 5G-AKA [36]	5898	10

5472 bits and 5898 bits n, respectively. Therefore, these approaches use cost factors that can influence the design and effectiveness of a protocol for a particular security solution.

Both the analytical and simulation evaluation approaches show similar outputs, indicating that SAP-AKA and NS-FId security protocols have better performance margins in terms of efficiency and computational cost but SAP-AKA has better throughput than both NS-FId and 5G-AKA protocols. The simulation of these protocols, in NS-3, allows us to compare the results and analysis from analytical modelling. Hence, it has been evidently shown that an analytical approach can effectively be used to evaluate a security protocol's performance. This can easily be achieved by integrating formal verification methods with analytical/mathematical performance evaluation techniques such as protocol symbolic/computation modelling and Markov Chain model [40].

7. Conclusion

During security protocols development and evaluation, security properties can be assessed through threat modelling or formal methods, and performance analysis can be conducted via mathematical modelling or simulations, these approaches have their limitations. We used *pi* calculus and Markov Chain processes to evaluate the performance of 5G security protocols based on the semantics of the protocols' specifications in ProVerif. We employed enhanced operational semantics, bisimilarity labelled to related rates of the processes. By looking at the enhanced labels and acquiring the actual values by providing more information on the architecture and cryptographic schemes in relation to the system model. This approach enabled the use of cryptographic primitives, mathematical techniques, computer states and processes to evaluate communication costs using message exchange, and protocol computational times were measured using an estimation of computation time. We were particularly interested in the quantitative

measurement of cryptography primitives and schemes of security protocols, allowing the protocols. Each cryptographic scheme has a different cost associated with it, which is determined by how it uses resources and time. In addition, the target cryptographic scheme and system model have an impact on the algorithm behaviour and cost, and vice versa. This work aligns with the evolution of computing and advances state of the art such as 5G in the context of advanced communications technologies, which requires the use of computation and mathematical techniques to solve state and process-based problems. Therefore, our approach makes the critical cost factors evident and improves protocols development, and the selection of efficient solutions. By providing a comprehensive and quantitative assessment of protocol performance, taking into account security properties and cryptographic costs. These methods can be applied to other communication systems besides IoT and mobile networks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

References

[1] C. Wang, Y. Zhang, X. Chen, K. Liang, Z. Wang, SDN-based handover authentication scheme for mobile edge computing in cyber-physical systems, IEEE Internet Things J. 6 (5) (2019) 8692–8701, Publisher: IEEE.

[2] S. Rostampour, M. Safkhani, Y. Bendavid, N. Bagheri, ECCbAP: A secure ECC-based authentication protocol for IoT edge devices, Pervasive Mob. Comput. 67 (2020) 101194, Publisher: Elsevier.

[3] Y. Zhang, B. Li, B. Liu, Y. Hu, H. Zheng, A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain, IEEE Internet Things J. 8 (18) (2021) 13958–13974, Publisher: IEEE.

[4] M. Vivekanandan, S.R. U, BIDAPSCA5G: Blockchain based Internet of Things (IoT) device to device authentication protocol for smart city applications using 5G technology, Peer-to-Peer Netw. Appl. 14 (2021) 403–419, Publisher: Springer.

[5] D. Basin, J. Dreier, L. Hirschi, S. Radomirović, R. Sasse, V. Stettler, A formal analysis of 5G authentication, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 1383–1396, Journal Abbreviation: CCS '18.

[6] E.K.K. Edris, M. Aiash, J. Loo, Formal verification and analysis of primary authentication based on 5G-AKA protocol, in: The Third International Symposium on 5G Emerging Technologies (5GET 2020), IEEE, Paris, France, 2020, pp. 256–261.

[7] J. Zhang, L. Yang, W. Cao, Q. Wang, Formal analysis of 5G EAP-TLS authentication protocol using ProVerif, IEEE Access (2020).

[8] E.K.K. Edris, M. Aiash, J. Loo, Network service federated identity (NS-FId) protocol for service authorization in 5G network, in: 5th IEEE International Conference on Fog and Mobile Edge Computing (FMEC 2020), IEEE, Paris, France, 2020.

[9] E.K.K. Edris, M. Aiash, J. Loo, M.S. Alhakeem, Formal verification of secondary authentication protocol for 5G secondary authentication, Int. J. Secur. Netw. 16 (4) (2021) 223–234.

[10] E. Ahmed, A. Naveed, S.H.A. Hamid, A. Gani, K. Salah, Formal analysis of seamless application execution in mobile cloud computing, J. Supercomput. 73 (10) (2017) 4466–4492.

[11] E.K.K. Edris, M. Aiash, ZKPVM: A zero-knowledge authentication protocol for VMs' live migration in mobile cloud computing, in: 13th International Conference on Software Technologies, ICSOFT, Porto, Portugal, 2018.

[12] A.M. Zbrzezny, A. Zbrzezny, S. Szymoniak, O. Siedlecka-Lamch, M. Kurkowski, Versectis-an agent based model checker for security protocols, in: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, 2020, pp. 2123–2125.

[13] S. Szymoniak, Security protocols analysis including various time parameters, Math. Biosci. Eng. 18 (2) (2021) 1136–1153, Publisher: AIMS Press.

[14] C. Cremers, C. Fontaine, C. Jacomme, A logic and an interactive prover for the computational post-quantum security of protocols, in: 2022 IEEE Symposium on Security and Privacy (SP), IEEE, 2022, pp. 125–141.

[15] Nsnam, ns-3 a discrete-event network simulator for internet systems, 2021, Issue: NS-3.33.

[16] A. Varga, Modeling and Tools for Network Simulation, Springer, 2010, pp. 35–59, Section: OMNeT++.

[17] MATLAB, MATLAB (R2019a), The MathWorks Inc., Natick, Massachusetts, 2019.

[18] W.J. Stewart, Introduction To the Numerical Solution of Markov Chains, Princeton University Press, 1994.

[19] B. Blanchet, B. Smyth, V. Cheval, M. Sylvestre, ProVerif 2.04: automatic cryptographic protocol verifier, user manual and tutorial, 2021, URL https://prosecco.gforge.inria.fr/personal/bblanche/proverif/, (Accessed on 02.08.2022).

[20] C. Bodei, M. Curti, P. Degano, M. Buchholtz, F. Nielson, H.R. Nielson, C. Priami, Performance evaluation of security protocols specified in LySa, Electron. Notes Theor. Comput. Sci. 112 (2005) 167–189.

[21] M. Abadi, B. Blanchet, C. Fournet, The applied Pi calculus: Mobile values, new names, and secure communication, J. ACM 65 (1) (2017) 1–41.

[22] C. Nottegar, C. Priami, P. Degano, Performance evaluation of mobile processes via abstract machines, IEEE Trans. Softw. Eng. 27 (10) (2001) 867–889.

[23] A. Koutsos, The 5G-AKA authentication protocol privacy, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2019, pp. 464–479.

[24] Z. Haddad, M.M. Fouda, M. Mahmoud, M. Abdallah, Blockchain-based authentication for 5G networks, in: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), IEEE, 2020, pp. 189–194.

[25] S. Gupta, B.L. Parne, N.S. Chaudhari, A generic construction for efficient and secure AKA protocol in 5G network, in: 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2018, pp. 1–6.

[26] M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: The spi calculus, Inf. Comput. 148 (1) (1999) 1–70.

[27] J. Mo, Performance modeling of communication networks with Markov chains, Synth. Lect. Data Manag. 3 (1) (2010) 1–90.

[28] M.D. Ryan, B. Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Vol. 5, 2011, pp. 112–142.

[29] E.K.K. Edris, M. Aiash, J. Loo, Formal verification of authentication and service authorization protocols in 5G enabled device-to-device communications using ProVerif, Electronics 10 (13) (2021) 1608.

[30] K. Bhargavan, B. Blanchet, N. Kobeissi, Verified models and reference implementations for the TLS 1.3 standard candidate, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 483–502.

[31] B. Blanchet, Automatic verification of correspondences for security protocols, J. Comput. Secur. 17 (4) (2009) 363–434.

[32] B. Blanchet, Modeling and verifying security protocols with the applied Pi calculus and ProVerif, Found. Trends Priv. Secur. 1 (1–2) (2016) 1–135,

[33] J. Hillston, A Compositional Approach To Performance Modelling, Vol. 12, Cambridge University Press, 2005.

[34] C. Bodei, M. Curti, P. Degano, M. Buchholtz, F. Nielson, H.R. Nielson, C. Priami, On evaluating the performance of security protocols, in: International Conference on Parallel Computing Technologies, Springer, 2005, pp. 1–15.
[35] A. Hodjat, I. Verbauwhede, The energy cost of secrets in ad-hoc networks (short paper), in: Proc. IEEE Circuits and Systems Workshop (CAS), Citeseer, 2002.
[36] 3GPP, Security Architecture; Procedures for 5G System, Technical specification (TS) 3GPP TS 33.501 V17.4.1 (2022-01), Third Generation Partnership Project, 2022, URL https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169.
[37] C. Priami, Stochastic pi-calculus with general distributions, in: Proc. of the 4th Workshop on Process Algebras and Performance Modelling (PAPM'96), Citeseer, 1996, pp. 41–57.
[38] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, M. Zorzi, End-to-end simulation of 5G mmWave networks, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2237–2263.
[39] S. Banerjee, V. Odelu, A.K. Das, S. Chattopadhyay, Y. Park, An efficient, anonymous and robust authentication scheme for smart home environments, Sensors 20 (4) (2020) 1215.
[40] E.K.K. Edris, M. Aiash, J. Loo, Formalization and evaluation of EAP-AKA' protocol for 5G network access security, Array (2022).