Quantitative Model Checking of an RSA-based Email Protocol on Mobile Devices

Sophia Petridou*, Stylianos Basagiannis*, Nikolaos Alexiou*, Georgios Papadimitriou* and Panagiotis Katsaros*
*Department of Informatics,
Aristotle University of Thessaloniki, Greece 54124
Email: spetrido, basags, nalexiou, gp, katsaros@csd.auth.gr

***Abstract*—The current proliferation of mobile devices has resulted in a large diversity of hardware specifications, each designed for different services and applications (e.g. cell phones, smart phones, PDAs). At the same time, e-mail message delivery has become a vital part of everyday communications. This article provides a cost-aware study of an RSA-based e-mail protocol executed upon the widely used Apple iPhone1,2 with ARM1176JZF-S, operating in an High Speed Downlink Packet Access (HSDPA) mobile environment. The proposed study employs formal analysis techniques, such as probabilistic model checking, and proceeds to a quantitative analysis of the e-mail protocol, taking into account computational parameters derived by the devices' specifications. The value of this study is to form a computer-aided framework which balances the tradeoff between gaining in security, using high-length RSA keys, and conserving CPU resources, due to hardware limitations of mobile devices. To the best of our knowledge, this is the first time that probabilistic model checking is utilized towards verifying a secure e-mail protocol under hardware constrains. In fact, the proposed analysis can be widely exploited by protocol designers in order to verify their products in conjunction with specific mobile devices.**

***Keywords*-Certified e-mail delivery; mobile devices; probabilistic model checking; quantitative analysis; RSA cryptosystem.**

I. Introduction

Nowadays, the widespread use of mobile communications [1], [2], [3], [4] along with the services and applications (e.g. cell phones, smart phones, PDAs) supported by the new generation's mobile devices, entails low-cost infrastructure executing high-consuming protocols [5], [6]. Given that certified e-mail message delivery is one of the most dominant applications used by mobile devices [5], it is fundamental to study the conditions under which a secure e-mail protocol is suitable for mobile devices. Formal analysis techniques, such as probabilistic model checking, are considered to be an effective way for studying security properties of communication protocols [7].

In this paper, we use the probabilistic model checker PRISM [8] to formally analyze the Certified Email Message Protocol Delivery (CEMD) [9], an RSA-based e-mail protocol. The proposed quantitative analysis is considered to preserve the security properties, i.e., fairness, timeliness, confidentiality and TTP invisibility, of CEMD. The CEMD protocol is modeled as a Continuous-Time Markov Chain (CTMC) [10], while its properties are expressed as Continuous Stochastic Logic (CSL) formulaes [11]. PRISM model checker [8] performs automated analysis of the CEMD protocol and verifies the security guarantees it provides. Then, the aforementioned CTMC model is used for the quantitative analysis of the protocol's computational cost.

Computational cost is considered in line with the CPU cycles and the time consumed by a mobile processor to perform RSA operations, i.e. encryption and decryption operations, required by the CEMD protocol. The current paper considers the widely used Apple iPhone1,2 with ARM1176JZF-S operating at $412MHz$ [12]. This actually means that the proposed computational cost analysis considers battery life through CPU cycles. Although display and applications, e.g., camera, mp3 and games, consume a great portion of battery life, it is found that CPU and memory are the dominant consuming subsystems in 3G mobile devices [5]. In fact, the CPU power is consumed due to instructions' execution and their fetching from the memories or caches. This in conjunction with the fact that CEMD protocol is analyzed for different key lengths ($128 - 2048$ bits) results in great deal of CPU expenditure.

Thus, experiments are launched on Apple iPhone1,2 operating in a High Speed Downlink Packet Access (HSDPA) environment, where multiple participants execute parallel sessions with bit error rate (BER) defined at 10^{-3} [13]. Mutual authentication of participants is based upon the well known RSA public key cryptosystem. Firstly, the 95% confidence interval of mean time required for RSA encryption and decryption operations, using keys of up to 2048 bits length, is calculated. Then, the CEMD protocol is modeled considering an initiator and up to 7 responders. The quantitative results provide the total amount of CPU cycles needed for the successful completion of CEMD session(s) for different key lengths in a finite amount of time. Probabilistic results are also derived in order to show the probability of initiated CEMD sessions being completed successfully.

A primitive model of this idea was firstly proposed by the authors in [14]. This early work employs the Texas Instruments TMS320C55x Family operating at $200MHz$. The novelty of the proposed extended model is that it exploits experimental results over the Apple iPhone1,2 with ARM1176JZF-S operating at $412MHz$. These results define cost in line with hardware specifications i.e. actual CPU cycles needed for RSA encryption and decryption opera-

tions, instead of clear numbers used in the model described in [14]. Thus, the new method constitutes a resource-aware approach towards studying protocols' cost under hardware specifications.

The remainder of this paper is organized as follows. Section II provides a brief review of related studies, in order to point out the novelty of the proposed analysis. Section III is an introduction to the probabilistic model checking principles based on CTMC models and CSL logic. The CEMD protocol and the developed CTMC model are presented in Section IV. Section V discusses the results derived from the quantitative analysis and their impact. Conclusions are given in Section VI.

II. Related Work

Nowadays, providing security services in mobile communications and verifying their properties is no longer optional due to their exponential growth [15]. More specifically, it is essential for protocols' designers to verify security and performance properties [16] of their products, as well as to quantify their cost-related properties. This fact makes probabilistic model checking a promising approach towards quantitative analysis of protocols [8], [17]. The importance of enabling quantitative analysis for a given cryptographic protocol was first shown in [18]. In that work, the author proposes a formal framework for weighting the cost of the participants, in order to verify the degree of a protocol's resistance against Denial of Service (DoS) attacks, in the context of the resource intensive task of mutual authentication. Recently, the aforementioned approach [18] has formed the basis for the analysis framework of [19]. The latter work [19] allows a more accurate representation of the protocol's computational cost. However, its drawback is that it employs simulation rather than verification for quantitatively analyzing the protocol's computational cost.

Another quantitative analysis approach is described in [20]. The authors specify a three-way-handshake of the TCP protocol in probabilistic rewriting logic in order to verify DoS resistance. The described representation generates a model for the VESTA toolset [21]. It is basically a timed probabilistic model that is analyzed by Monte Carlo simulation, upon which a series of interrelated statistical hypothesis tests are applied, in order to check whether the quantitative property of interest is fulfilled. However, such an approach, also known as statistical model checking, does not produce the same accurate results as the ones obtained by probabilistic model checking [22]. Moreover, the aforementioned approach is not appropriate for the analysis of communication protocols, since it does not cope with cost-related properties, such the ones incorporated in the proposed model.

III. Probabilistic Model Checking Analysis of CEMD

As it is stated in Section I, the CEMD protocol is modeled as a Continuous-Time Markov Chain (CTMC) [10]. A CTMC is a stochastic process that satisfies a Markov property, e.g. which is the total computational cost for completing a protocol's session in a finite amount of time? In a CTMC, the waiting time of a transition from a state i to a state j is governed by a negative exponential distribution, the parameter of which is *transition rate* q_{ij}. In our case, this exponential distribution depends on BER parameter, which influences the message exchanges, since 10^{-3} is a typical value of it in mobile environments [13]. CTMCs are widely used in protocols' analysis due to their strength in representing dynamic behavior, physical processes, and queueing systems with Poisson arrival rates [17]. Thus, developing a CTMC model for CEMD, we extract quantitative results based upon experiments run on a mobile device, e.g. Apple iPhone1,2, in order to verify the total amount of computational cost, e.g. CPU cycles, needed to achieve fair multiple email delivery in a HSDPA mobile environment. While CEMD is modeled as a CTMC, its properties are expressed as Continuous Stochastic Logic (CSL) formulaes [11] and verified over the full state space of the model.

A. Preliminaries

PRISM is a powerful probabilistic model checking framework for verifying the quantitative properties of a protocol [8]. In general, model checking involves the verification of properties over labeled state transition systems [23]. In PRISM, a probabilistic model is defined as a set of m reactive modules, $M = \{M_1, \ldots, M_m\}$. The CTMC model, developed in this work, consists of three (3) modules, namely $M = \{M_I, M_R, M_{TTP}\}$. Module M_I represents the protocol's initiator, while module M_R embodies a fixed set of responders R communicating with I, thus modeling a parallel sessions scenario. It is assumed that all the responders R_i, $i = 1, \ldots, 7$ trust the same TTP entity, represented by the module M_{TTP}. Each M_i module is defined as a pair of (Var_i, C_i), where Var_i is a set of integer-valued local variables with finite range and C_i is a set of commands.

The behavior of module M_i is defined by the set of commands C_i. Each command $c \in C_i$ takes the form of $(g, (\lambda_1, u_1), \ldots, (\lambda_{n_c}, u_{n_c}))$, comprising a guard g and a set of pairs (λ_j, u_j), where $\lambda_j \in \Re_{>0}$ and u_j is an update for each $1 \leq j \leq n_c$. A guard g is a predicate over the set of all local variables Var and each update u_j corresponds to a possible transition of module M_i. If Var_i contains n_i local variables, $\{v_1, \ldots, v_{n_i}\}$, then an update takes the form $(v'_1 = expr_1) \cap \ldots \cap (v'_{n_i} = expr_{n_i})$, where $expr_j$ is an expression in terms of the variables in Var. If an update leaves the values of some variables in Var_i unchanged, the

model description may omit this information. The constants λ_j determine the rates attached to the transitions [24]. In our model, λ_j depends on BER, since the rate of messages' delivery is affected by the transmission rate and BER.

All the above compose a CTMC model which is defined as a tuple $(S, \bar{s}, Rt, L)$, where: S is a finite set of states, $\bar{s} \in S$ is the initial state, $Rt : S \times S \rightarrow \Re_{\geq 0}$ is the transition rate matrix and $L : S \rightarrow 2^{AP}$ is the labeling function of atomic propositions AP that are true in S. Once the CTMC is constructed, its properties are encoded as CSL formulaes [11]. In CSL the analyst develops partial specifications of the steady-state and the transient behavior of CTMCs. The syntax of CSL is as follows:

$$\phi \ ::= \ true \mid \alpha \mid \phi \ \wedge \ \phi \mid \neg\phi \mid P_{\bowtie p}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi]$$

$$\psi \ ::= \ X \ \phi \mid \phi \ \mathcal{U}^{\leq t} \ \phi \mid \phi \ \mathcal{U} \ \phi$$

where a is an atomic proposition, operator $\bowtie \in [\leq, <, \geq, >]$, $p \in [0, 1]$ and $t \in \Re_{\geq 0}$. $P_{\bowtie p}[\psi]$ indicates that the probability of the path formula ψ, which is satisfied from a given state in a CTMC model, satisfies $\bowtie p$. CSL can also describe path formulaes with operators such as X (next) or $\mathcal{U}^{\leq t}$ (time bounded until with $t \in \Re_{\geq 0}$), as shown above. Finally, the $\mathcal{S}$ operator refers to the steady-state behavior of the CTMC. Formula $\mathcal{S}_{\bowtie p[\phi]}$ means that the steady-state probability of being in some state satisfying ϕ meets the bound $\bowtie p$.

For a CTMC $(S, \bar{s}, Rt, L)$, a reward structure is a tuple (ϱ, ι), where: $\varrho : S \rightarrow \Re_{\geq 0}$ is a vector of state rewards and $\iota : S \times S \rightarrow \Re_{\geq 0}$ is a matrix of transition rewards. A reward structure (ϱ, ι) for a CTMC allows the specification of four distinct types of rewards Rw, namely instantaneous, cumulative, reachability and steady-state rewards [7]. Cumulative reward properties are expressed by the formula $Rw_{\bowtie r}[\mathrm{C}^{\leq t}]$, which denotes that the expected reward cumulated up to time-instant t is $\bowtie r$, and employed for the quantitative verification of the proposed CTMC model.

IV. The fair Certified Email Message Delivery protocol - CEMD

The increased demand for reliable e-mail services in mobile communications led the researchers to propose secure e-mail protocols based upon strong security mechanisms for mutual authentication, such as the RSA cryptosystem [25]. In this study, we analyze the CEMD protocol, a fair Certified E-mail Delivery protocol [9]. When CEMD includes RSA cryptography it provides non-repudiation of origin, non-repudiation of receipt and strong fairness, whilst it makes use of an off-line and transparent trusted third party (TTP), in case that the communicating parties fail to complete the e-mail for receipt exchange due to network failures or a party's misbehavior. Besides the above, its design aimed at reducing the overall computational cost (i.e., cryptographic operations) for protocol's efficiency and cost-effectiveness [9].

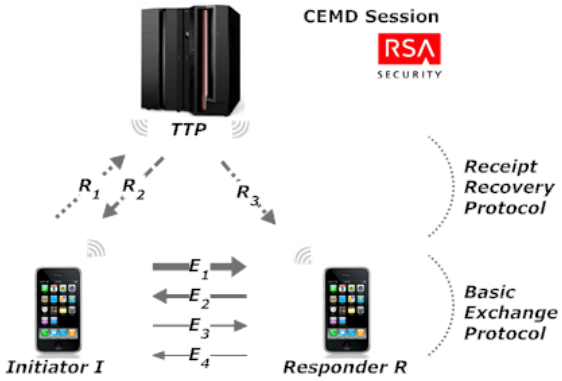


Figure 1. Representation of a single CEMD session

More specifically, the CEMD protocol comprises two sub-protocols, namely the Exchange Protocol and the Receipt Recovery Protocol, as shown in Fig. 1. The Exchange Protocol consists of the following four (4) discrete messages:

1) $E_1 = (h(M), Sign_I(h(M)))$: initiator I transfers to responder R a hash value $h(M)$ of the message M along with its digital signature $Sign_I(h(M))$ on M.
2) $E_2 = (VRES_R, Cert_R)$: responder R verifies $Sign_I(h(M))$ from message E_1 and confirms the result using $h(M)$. Then, he computes his Verifiable and Recoverable Encrypted Signature ($VRES_R$) message and sends to I a message E_2 consisting of $VRES_R$ and his certificate $Cert_R$, signed by TTP.
3) $E_3 = (M)$: message E_3 constitutes the e-mail message M that I wants to send to R.
4) $E_4 = (r_b)$: responder R generates and sends a random prime number r_R to initiator I in order that I is able to derive R's correct receipt using r_R.

The Receipt Recovery Protocol consists of three (3) discrete messages as follows:

1) $R_1 = (M, Cert_R, VRES_I)$: the initiator I transfers to the TTP the message R_1 consisting of the message M, the responder's R certificate $Cert_R$ and its $VRES_I$, where $VRES_I$ is derived from $VRES_R$ such that TTP accepts I's request.
2) $R_2 = (r_R)$: TTP sends to I the message R_2 containing the r_R in order that I computes the receipt of the responder R.
3) $R_3 = (M)$: TTP forwards the original message M to the responder R.

CEMD protocol has to provide certain security properties towards its participants, namely *fairness*, *timeliness*, *confidentiality* and *TTP invisibility* [9]. The proposed CTMC model of CEMD protocol is designed in respect to the aforementioned properties. CEMD participants are considered to employ an off-line TTP entity to ensure fair completion of the CEMD protocol, and, thus, neither the initiator nor the responder(s) do they gain advantage in a random

protocol's interruption. Then, we considered that all sessions will not fail once initiated. This design feature will force successful completion of each session in a finite amount of time allowing us to anticipate timeliness property in results. Confidentiality of the participants is kept, since RSA encryption (using modulo exponentiations) of all CEMD messages is considered unbreakable. Finally, in the proposed model the TTP entity involves in the CEMD session only when participant I fails to receive a proper receipt of R. Otherwise TTP does not interfere to the protocol resulting in TTP invisibility.

The aforementioned properties entails strong cryptographic operations [26], and thus, increased computational cost for the protocol's participants. It is therefore questionable whether a secure protocol, such as CEMD, can effectively operate over mobile environments characterized by low-cost infrastructure, such as mobile devices. Thus, the purpose of the proposed analysis is to provide quantitative results for the computational cost of the CEMD protocol. In line with this goal, we designed a CTMC model for the CEMD protocol. The Markov Chain created was augmented with cumulative reward properties Rw that represent the computational cost imposed when the model reaches a state, i.e., s.

A. The CTMC model of the RSA-based CEMD protocol

The CTMC model of the RSA-based CEMD protocol, designed in this work, consists of three (3) modules, namely $M = \{M_I, M_R, M_{TTP}\}$, as it has been mentioned in Section III-A. These modules represent all the necessary participants considered in a CEMD session. More specifically, modules M_I and M_{TTP} depict single CEMD entities, i.e., the initiator and TTP, respectively. Module M_R represent a fixed set of responders R communicating with I, modeling in this way a parallel session scenario. A number of up to 7 parallel sessions is assumed with all the R_i, $i = 1, \ldots, 7$ responders trust the same TTP entity. Nowadays, it is usual that a simple email user initiates a fair certified email session with more than one participants concurrently, but at the same time hardware limitations pose restrictions to a limited number of parallel sessions. In our model, apart from the TTP entity, the CEMD participants are Apple iPhone1,2 with ARM1176JZF-S at $412MHz$ [12] operating in a HSDPA mobile environment with BER defined at 10^{-3} [13].

Considering parallel sessions, the modules of set M interact by updating their local variables in line with the protocol's communication steps presented in Section IV. A number of these local control variables is defined to have a maximum value equal to $number\_of\_R\_machines$, which represents the number of parallel sessions. Then, variables' updates correspond to the modeled state transitions for the distinct participants. For example, $ACK\_counter\_I$ local control variable of M_I counts the initial ACK messages received by responders R. Obviously, once the initiator starts a session with R_i sending an ACK message, he is waiting for his response and does not continue to the next state until he receives it. In our model successful reception of ACK depends on BER. Thus, $ACK\_counter\_I \in Var_I$ is used in guard $g = (ACK\_counter\_I < number\_of\_R\_machines)$, which controls ACKs' reception. If g is TRUE then the update $(ACK\_counter\_I' = ACK\_counter\_I + 1)\ \&\ (I\_state' = 1)$ is performed with rate $\lambda_1 = 1 - ber$. Otherwise, $(I\_state' = 0)$ with rate $\lambda_2 = ber$. For the synchronization of M_I and M_R, synchronization labels are used, e.g., $[I\_send\_ACK]$.

Parameter $number\_of\_R\_machines$ is also used to control the TTP's visibility. TTP can either be visible or not during a session according to I's choice, as described in the CEMD principles [9]. If TTP is not involved in a CEMD session, then I and R complete their session exchanging messages E_1-E_4. On the contrary, if TTP is involved, messages $R1$-$R3$ will be also exchanged to assist fair completion of the initiated CEMD session. Variables $TTP\_certificates$ and no_TTP are associated with the TTP invisibility security property and count the sessions completed with TTP being visible or invisible, respectively. Then, a formula end, depicted the model's final state, is defined to be the sum of $TTP\_certificates$ and no_TTP that has to be equal to $number\_of\_R\_machines$.

Once the CEMD modules are constructed inside the CTMC model, computational cost parameters are derived according to the CEMD specifications [9]. In the proposed model, computational cost is calculated in line with the CPU cycles required for processing CEMD RSA-based messages, i.e., $E_1 - E_4$ and $R_1 - R_3$ (shown in Fig. 1). The dominant calculations through the above operations are RSA encryption and decryption performed by the CEMD participants, while other mathematical calculations, e.g. multiplication or division, are considered negligible. In order to estimate computational cost of RSA encryption and decryption, we launched a series of experiments on an Apple iPhone1,2 at $412MHz$ [12] and we recorded the time needed by this device to perform the RSA instructions. We developed the basic RSA cryptosystem in C and calculated the time in μsec for ASCII text encryption and decryption when different RSA keys are used (e.g., 128, 256, 512, 1024 and 2048 bits). The 95% confidence intervals of mean time required for RSA encryption and decryption for the above key lengths are shown on Tables I and II, respectively. At this point, it should be added that, traditionally, the TTP entity is not a mobile device. Thus, our model considers it to operate at 2GHz adjusting the CPU cycles respectively.

Finally, the CPU cycles, corresponding to the reported μsec, as well as the value of BER, defined at 10^{-3}, are embedded in the proposed CTMC in order to build reward structures Rw, one for each RSA key length case. In our model, cumulative rewards, expressed by the formula

Table I
95% CONFIDENCE INTERVALS OF TIME FOR ENCRYPTION PROCESS ON APPLE IPHONE1,2 (412MHZ)

key (bits)	time (μsec)	CPU Mcycles
128	555.24 ± 9.77	0.23
256	759.56 ± 10.11	0.32
512	1327.03 ± 15.25	0.55
1024	3587.69 ± 123.31	1.49
2048	10718.54 ± 654.47	4.47

Table II
95% CONFIDENCE INTERVALS OF TIME FOR DECRYPTION PROCESS ON APPLE IPHONE1,2 (412MHZ)

key (bits)	time (μsec)	CPU Mcycles
128	3415.94 ± 62.47	1.42
256	5612.79 ± 74.42	2.34
512	15045.19 ± 169.37	6.27
1024	73514.45 ± 807.99	30.63
2048	363360.61 ± 7291.11	151.40

$Rw_{\bowtie r}[\mathrm{C}^{\leq t}]$ (as mentioned in Section III-A), are employed, since we are interested in calculating the total computation cost of all the initiated sessions up to a finite amount of time t.

V. EXPERIMENTAL RESULTS

This section presents the quantitative verification results derived from the proposed CTMC model of the RSA-based CEMD protocol. These results depict the computational cost of an Apple iPhone1,2 operating at $412MHz$ [12] mobile device when executing the CEMD protocol. As it has been mentioned, it is considered that all M_I and M_R modules are iPhone1,2 devices in a HSDPA mobile environment, where multiple responders R_i, $i = 1, \ldots, 7$, execute parallel CEMD sessions with the initiator I, supposing that the BER has the typical value of 10^{-3} [13]. In Section IV-A, Tables I and II present the CPU cycles and the time consumed by the aforementioned mobile processor to perform RSA operations, i.e. encryption and decryption operations, for different key lengths. This information is incorporated in cumulative rewards in order to launch a series of queries.

For the following queries, cumulative reward properties of the form $Rw_{\bowtie r}[\mathrm{C}^{\leq t}]$ are defined, as described in Section IV-A. Cumulative reward properties associate a reward with each path of the model, but only up to a given time boundary. The property $[\mathrm{C}^{\leq t}]$ corresponds to the reward cumulated along a path, until t time units have elapsed. Timeliness, as a security property that an e-mail protocol should provide, can be verified through a user defined time boundary. The first CSL query is defined as a cumulative type query as follows:

$$Query1 : Rw\{\text{“}Computational\_Cost\text{”} \ \forall \text{RSA key}\} =? \\ [C <= C_0], R_i, i = \{1, \ldots, 7\}, C_0 = 100$$

The explanation of the above query is: “which is the overall

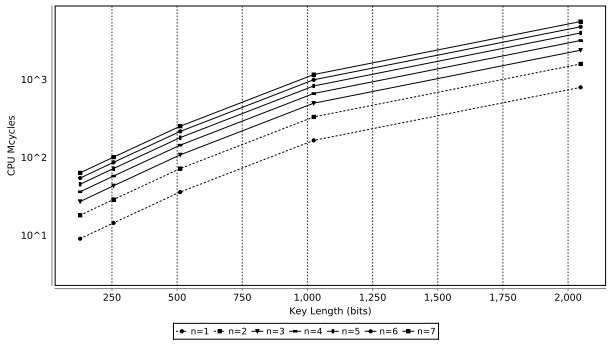


Figure 2. Cumulated computational cost as a function of RSA key lengths for $n = 1, \ldots, 7$ responders

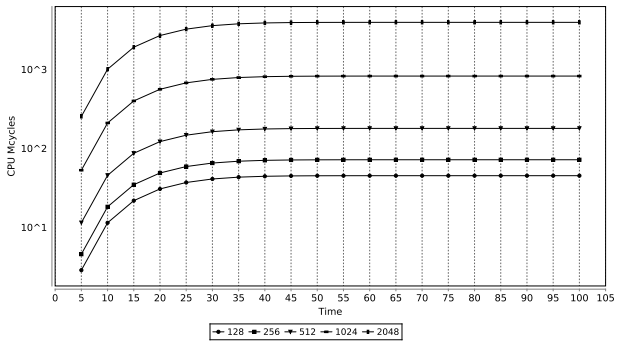


Figure 3. Cumulated computational cost as a function of time for $n = 5$ responders and for different RSA key lengths

$Computational\_Cost$ for completing (all) protocol's session(s) in a finite amount of time C_0?”.

Fig. 2 depicts the cumulated computational cost, expressed in CPU Mcycles, as a function of key length when the M_I initiator starts up to $n = 1, \ldots, 7$ sessions with the responders $\{R_1, \ldots, R_7\}$. Each point in this graph is derived when $C_0 = 100$ time units. The ascending trend of each curve indicates that, as the RSA key length increases the corresponding computational cost for a given number of parallel sessions is also increased. At the same time, the allocation of curves from $n = 1$ to 7 provides evidence that for a given value of key length the more the parallel sessions the higher the computational cost. An interesting observation is that the cost for one single session with RSA key at 2048 bits is 795 Mcycles, which is more than three times greater compared to the corresponding cost, i.e., 252 Mcyles, for 7 parallel sessions using RSA key at 512 bits.

In $Query1$, we consider that the TTP entity is involved in the protocol with a probability equal to 0.5. $Query2$, that follows, also defines that $P_{TTP} = 0.5$. However, while $Query1$ provides the cumulated computational cost when C_0 reaches at 100 time units, $Query2$ depicts the cost as a function of time, expressed in time units, for up to 100 time units. For the analysis purpose, we define a CSL query as

follows:

$$Query2 : Rw\{\text{“}Computational\_Cost\text{”}\ \forall\text{RSA keys}\} =? \\ [C <= C_0], R_i, i = 5, P_{TTP} = 0.5$$

The explanation of the above query is: “which is the overall $Computational\_Cost$ for completing five protocols' sessions, i.e. R_5, in a finite amount of time C_0, with $P_{TTP} = 0.5$ and for different values of RSA key lengths?”.

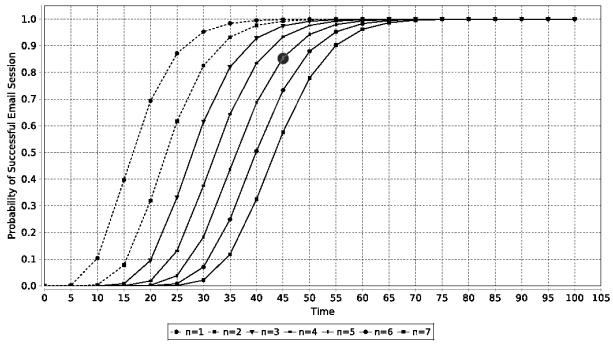
Fig. 3 provides the cumulated cost in Mcycles for $n = 5$ responders as a function of time for different RSA key lengths. We firstly observe that after a finite amount of time, computational cost remains unchanged confirming the timeliness property. In that period, the protocol will no longer need additional CPU resources, and thus, it is speculated that all the 5 CEMD sessions will be finished. The curves in this figure conform to the anticipated observations, i.e. for a given number of parallel sessions it is natural that the increase in RSA key length overheads the CEMD's execution by the mobile device. However, overheads behave non-linearly compared to the selected key lengths. For example, a 50% increase to the key length from 1024 to 2048 bits leads to an 80% cost overhead from $8.2 * 10^2$ Mcycles to $3.9 * 10^3$ Mcycles.

Apart from quantitative results, one of the benefits of using CTMC modeling through probabilistic model checking is the capability of the analysis to produce probabilistic results about the actual probability of certain events to occur. In this context, it is desired to measure the probability of all the initiated sessions to be completed in a certain amount of time. Thus, we define the following CSL query:

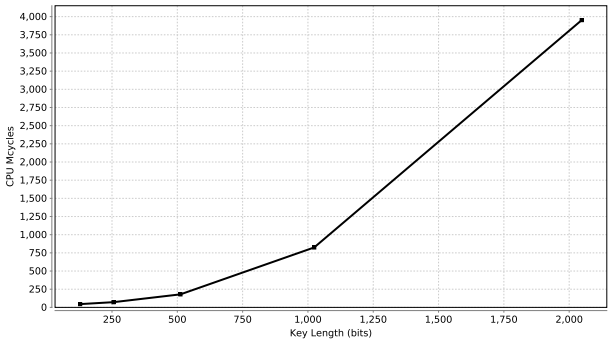
$$Query3 : P =? \ [F <= C_0 \ end], R_i, i = \{1, \ldots, 7\}, \\ C_0 = 100, P_{TTP} = 0.5$$

The explanation of the above query is: “with which probability all initiated sessions will be completed in a time boundary C_0?”. For the defined property of $Query3$, the verification process will search all the produced state space in order to find final states before C_0 time units, for which the formula “end” will be true. In our model, the formula “end” represents a boolean expression which controls that all the initiated sessions will eventually completed successfully. As shown in Fig. 4(a), all the initiated sessions will eventually finish after a finite amount of time. Obviously, probability $P \simeq 1$ for one session $n = 1$ sooner, e.g., after 40 time units, than for seven $n = 7$ parallel sessions, e.g., after 70 time units.

As shown in Fig. 4(a), the calculated probability for $n = 5$ parallel sessions of completing the CEMD sessions is $P = 0.856$ after 45 time units. For this time boundary, it is interesting to calculate the computational cost when different RSA keys are used. Thus, we launched quantitative results for specific time units and for a fixed number of parallel



(a) The probability for $n = 1, \ldots, 7$ responders to complete their sessions as a function of time



(b) Cumulated computational cost as a function of RSA key length for $n = 5$ responders, when $C_0 = 45$ time units and $P = 0.85$ of completing CEMD session

Figure 4. Combining probabilistic and computational cost results

sessions. The query:

$$Query4 : Rw\{\text{“}Computational\_Cost\text{”}\ \forall\text{RSA keys}\} =? \\ [C <= 45], R_i, i = 5$$

can be explained as: “which is the overall $Computational\_Cost$ for $n = 5$ protocols' sessions in a finite amount of time equal to 45 time units for different RSA key lengths?”. For this time boundary, computational cost depicted in Fig. 4(b). For example, using a 128 bit key the CPU Mcycles consumed will be at 45.1. $Query3$ and $Query4$ can be appropriately used in order to combine probabilistic and computational cost results.

VI. Conclusions and Future Work

In this paper we introduce quantitative verification using probabilistic model checking as a means for the evaluation of an RSA-based e-mail protocol executed upon the widely used Apple iPhone1,2 in a HSDPA mobile environment. Since the analyzed CEMD protocol uses the RSA public key cryptosystem, the 95% confidence interval of mean time required for RSA decryption and encryption operations, using keys of up to 2048 bits length, is calculated. We proceeded to our analysis designing a parameterized CTMC

model which incorporates computational cost parameters derived by hardware specifications and considers up to 7 parallel sessions. As a result, the analysis produces valuable quantitative outputs about the actual computational cost expressed in CPU cycles for the successful completion of CEMD session(s) as a function of RSA key length or time units. As a future work, we aim at enhancing the proposed quantitative verification analysis with energy consumption issues, such as battery life, as well as with other mobile environments' parameters, such as BER and bandwidth parameters.

REFERENCES

[1] D. Miorandi, E. Uhlemann, S. Vitturi, and A. Willig, "Guest editorial: Special section on wireless technologies in factory and industrial automation, part i," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 95–98, 2007.

[2] Q. Bi, G. Zysman, and H. Menkes, "Wireless mobile communications at the start of the 21st century," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 110–116, 2001.

[3] Y. Zhang, N. Ansari, and H. Tsunoda, "Wireless telemedicine services over integrated ieee 802.11/wlan and ieee 802.16/wimax networks," *IEEE Communications Magazine*, vol. 17, no. 1, pp. 30–36, 2010.

[4] C. Liaskos, S. Petridou, and G. Papadimitriou, "Cost-aware wireless data broadcasting," *IEEE Transactions on Broadcasting*, vol. 56, no. 1, pp. 66–76, 2010.

[5] N. Sklavos and K. Touliou, "A system-level analysis of power consumption & optimizations in 3g mobile devices," in *Proc. of the 1st International Conference on New Technologies, Mobility & Security (NTMS'07)*, France, May 2007, p. 225235.

[6] N. Sklavos and Z. Xinmiao, *Wireless Security and Cryptography: Specifications and Implementations*. FL, USA: CRC Press, Inc., 2007.

[7] M. Z. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, Italy, May 2007, pp. 220–270.

[8] ——, "Prism 2.0: A tool for probabilistic model checking," in *Proc. of the 1st Int. Conf. on Quantitative Evaluation of Systems (QEST'04)*, Netherlands, Sep. 2004, pp. 322–323.

[9] A. Nenadic, N. Zhang, and S. Barton, "Fair certified e-mail delivery," in *Proc. of the 2004 ACM Symposium on Applied Computing (SAC'04)*, USA, Mar. 2004, pp. 391–396.

[10] W. J. Stewart, *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.

[11] A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton, "Model-checking continous-time markov chains," *ACM Transactions on Computational Logic*, vol. 1, no. 1, pp. 162–170, 2000.

[12] "Apple-iphone 3g," World Wide Web electronic publication. [Online]. Available: http://www.apple.com/pr/library/2010/10/18results.html

[13] N. K. C. K. Sohaib and J. Nordberg, "Hsdpa system simulation," in *Proc. of the 2nd International Symposium on Communications, Control and Signal Processing (ISCCSP'06)*, Morocco, Mar. 2006.

[14] S. Basagiannis, S. Petridou, N. Alexiou, G. Papadimitriou, and P. Katsaros, "Quantitative analysis of a certified e-mail protocol in mobile environments: A probabilistic model checking approach," *to appear in Computers & Security, Elsevier*, 2011.

[15] P. Prasithsangaree and P. Krishnamurthy, "On a framework for energy-efficient security protocols in wireless networks," *Computer Comm.*, vol. 27, no. 17, pp. 1716–1729, 2004.

[16] S. Basagiannis, P. Katsaros, and A. Pombortsis, "Intrusion attack tactics for the model checking of e-commerce security guarantees," in *SAFECOMP*, Germany, sep 2007, pp. 238–251.

[17] S. Basagiannis, P. Katsaros, A. Pombortsis, and N. Alexiou, "Probabilistic model checking for the quantification of dos security threats," *Computers & Security*, vol. 28, no. 6, pp. 450–465, 2009.

[18] C. Meadows, "A cost-based framework for analysis of denial of service in networks," *Journal of Computer Security*, vol. 9, no. 1-2, pp. 143–164, 2001.

[19] S. Tritilanunt, C. Boyd, E. Foo, and J. M. G. Neto, "Using coloured petri nets to simulate dos-resistant protocols," in *Proc. 7th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Denmark, October 2006, pp. 261–280.

[20] G. Agha, M. Greenwald, C. A. Gunter, S. Khanna, J. Meseguer, K. Sen, and P. Thati, "Formal modeling and analysis of dos using probabilistic rewrite theories," in *Proc. the IEEE Workshop on Foundations of Computer Security (FCS'05)*, Chicago, IL, USA, June 2005.

[21] K. Sen, M. Viswanathan, and G. Agha, "Vesta: A statistical model-checker and analyzer for probabilistic systems," in *Int. Conference on Quantitative Evaluation of Systems*, Los Alamitos, CA, USA, 2005, pp. 251–252.

[22] ——, "On statistical model checking of stochastic systems," in *Proc. of the 17th Int. Conference on Computer Aided Verification (CAV'05)*, UK, July 2005, pp. 266–280.

[23] E. Clarke, O. Grumberg, and D. Peled, *Model Checking*. The MIT Press, 2000.

[24] A. Bianco and L. de Alfaro, "Model checking of probabilistic and nondeterministic systems," in *Proc. of the 15th Conference on Foundations of Computer Technology and Theoretical Computer Science*, India, December 1995, pp. 499–513.

[25] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[26] G. Ateniese and C. Nita-Rotaru, "Stateless-recipient certified e-mail system based on verifiable encryption," in *Proc. of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology (CT-RSA'02)*, Germany, February 2002, pp. 182–199.