Security and Privacy in the Internet-of-Things Under Time-and-Budget-Limited Adversary Model

Sriram N. Premnath and Zygmunt J. Haas, *Fellow, IEEE*

***Abstract*—Internet-of-Things (IoT) represents an emerging era of networking that connects a variety of common appliances to one another, as well as with the rest of the Internet, to vastly improve our lives. Despite being significantly resource-constrained, IoT nodes are expected to participate in numerous computationally-intensive security protocols to overcome threats from the public Internet. Given that IoT nodes (e.g., smart meters) typically exchange tactical data that requires data protection for a short time span of up to a few days, we examine the use of smaller cryptographic key sizes to provide IoT security. We show that small key sizes quite drastically reduce the cryptographic computational processing requirements for IoT nodes. We estimate the cost of breaking public key crypto systems when the adversary is limited by the available resources (i.e., *dollar cost*) and time (i.e., *number of days*). We consider Moore's law, as well as *More than Moore*, and *Less than Moore* technology growth rates, in conjunction with the capabilities of a real-world key-breaker to calculate the cost estimates. Finally, we also present the trade-off between the processing load for an IoT node versus the desired time span of privacy protection.**

***Index Terms*—Internet of things, cryptographic key size, smart grid.**

I. Introduction

INTERNET of things (IoT) represents an emerging era of networking that connects a variety of day-to-day, commonly-used, home appliances to one another, as well as with the rest of the Internet, with the goal of vastly improving our lives. IoT enables numerous, useful, real-world applications, such as assisted-driving, tele-healthcare, inventory tracking, smart homes/offices, etc. However, successful adoption of IoT depends greatly on its ability to provide sufficient protection of the users and their private data against the Internet real-time security threats ([1], [2]).

To provide security and privacy, as well as to assimilate with the rest of the Internet, IoT nodes are required to participate in a variety of widely-used and well-established security mechanisms (e.g., SSL, IPSec, PKI, Diffie-Hellman key exchange, etc.) that require performing numerous computationally intensive cryptographic operations. However, a typical node in the IoT is far more resource-constrained than a host in today's Internet. Consequently, achieving security and privacy in IoT presents a major challenge.

Considering that IoT nodes typically exchange *tactical* data, as opposed to *strategic* data, it is more desirable to achieve data protection for a time span ranging from *real-time to a few days*, as opposed to several years or decades. Therefore, we examine the use of public key crypto systems with smaller key sizes, which results in computationally vastly less-intensive cryptographic operations.

To further clarify the goal of our work, consider the following example of a *smart grid network*, in which a *smart meter* sends energy usage information to a *utility company server* for billing purposes, real-time power grid optimization, etc. Now, assume that the house owner is out of town for 4–5 days. If a remote adversary eavesdrops on the communication between the smart meter and the utility company server and manages to recover the information regarding the reduction in power usage, then the adversary learns that the house could be empty, and may consider it as a target for burglary ([10]).

In the above example, it is necessary to encrypt the information exchange between the smart meter and the utility company server. The energy usage information is useful to the adversary if it can recover that information within a few days before the house owner returns. In such a scenario, it is fairly reasonable to encrypt the information using substantially smaller key sizes than the typical 128-bit symmetric key size, which is recommended for protecting information for a long period of about 30 years ([3], [5]). In this work, we determine the cost for an adversary to break the system in terms of number of days and dollar budget for various small key sizes.

We summarize the main contributions of our work as follows. First, given that IoT data typically requires protection for only a few days, we introduce a more appropriate *time-and-budget limited adversary model* suitable for IoT. Second, using the capabilities of an efficient real-world key-breaker based on FPGA, in conjunction with Moore's law, we estimate the cost of breaking public-key crypto systems under small key sizes, in terms of the number of days and the associated dollar cost. Third, we show that small cryptographic keys very drastically reduce the cryptographic computational processing requirements for IoT nodes. Finally, we also present the trade-off between the processing load for an IoT node versus the desired time span of privacy protection.

II. Adversarial Model

We consider the following adversarial model in our evaluation of the security and privacy of data in IoT.

Manuscript received October 29, 2014; accepted February 24, 2015. Date of publication March 4, 2015; date of current version June 18, 2015. This work was part of the National Science Foundation (NSF) Future Internet Architecture Project, and was supported by the NSF under the following grant numbers: CNS-1040689, ECCS-1308208, and CNS-1352880. The associate editor coordinating the review of this paper and approving it for publication was X. Fu.

S. N. Premnath is with Cornell University, Ithaca, NY-14853 USA (e-mail: sriram.np@cornell.edu).

Z. J. Haas is with Cornell University, Ithaca, NY 14853 USA and also with University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: haas@ece.cornell.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LWC.2015.2408609

Budget-limited adversary: The adversary has a limited budget in terms of *Dollar amount*; for example, \$10,000 only. Alternatively, the monetary value of the information being protected is at least as much as the Dollar budget of the adversary, which may serve as a motivation for the adversary to break the crypto system (also refer our Smartgrid example in Section IV). In our example above, an adversary with a budget limit of \$10,000 may consider breaking the crypto system only if he/she gains far more than \$10,000.

Time-limited adversary: The adversary has a limited amount of time to break the system; for example, the computational resources to break the crypto system may be available to the adversary for a time period of one week. Alternatively, the data being protected is of useful value to the adversary only for a prescribed amount of time; for example, if the protected data is recovered within the time span of a few days.

Our time-and-budget limited adversarial model more accurately captures the requirements of certain classes of IoT applications.

III. Motivation: Reduction in Computational Requirements for IoT Nodes

We motivate our rationale for why it is acceptable to use small cryptographic keys in certain situations.

In some widely-used protocols such as SSL, for each new session between a pair of parties, a new shared secret key is established using the Diffie-Hellman key exchange (DHKE) protocol using expensive cryptographic computational operations. The *ephemeral* shared secrets established in this manner are not derived from long term private keys used for the purpose of server authentication, digital signatures, etc. Using DHKE in conjunction with SSL provides *forward secrecy*, i.e., even if the long term private key becomes compromised in the future, the data exchanged in the past sessions still remains protected ([11], [12]).

When the DHKE parameters are considerably small, it results in a drastic reduction in the computational processing power required for cryptographic operations, which is very desirable for IoT nodes.

A small number of modular exponentiation operations are performed during the DHKE protocol (as well as in other well-known cryptographic algorithms such as RSA, ElGamal encryption/decryption, etc.). Modular exponentiation operation is carried out efficiently through repeated squaring and reduction ([13]). Let n denote the number of bits representing the modulus value. Then, the computational complexity of the modular exponentiation, expressed in terms of the number of bit operations, is $O(n^3)$ ([13]).

To compare the reduction in processing when using smaller key sizes, we define *relative computational effort* as the ratio, $Effort_{rel}(n) = (\frac{n^3}{3248^3})$; note that a 3248-bit asymmetric/public key modulus value is equivalent to a symmetric/secret key of size 128 bits [5].

For example, when a 1024-bit modulus is used, the relative computational effort equals $Effort_{rel}(1024) = 3.1\%$; in other words, the number of bit operations when using a 1024-bit modulus is only about 3.1% of the total number of bit operations carried out when a 3248-bit modulus were instead used. Fig. 1 depicts the relative computational effort for modular exponentiation under various modulus sizes.

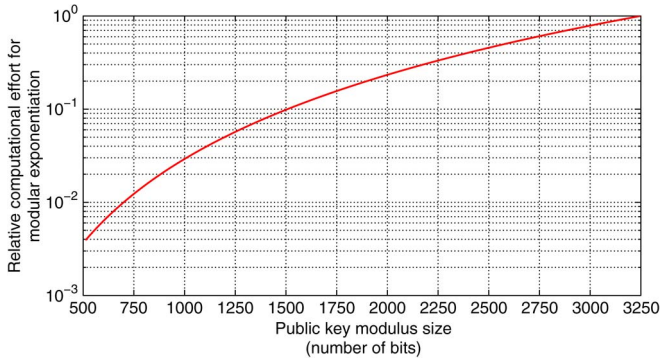

Fig. 1. Relative computational effort for modular exponentiation operation versus public key modulus size.

Our result in Fig. 1 implies that, where permissible, use of smaller key sizes results in significantly faster cryptographic operations, and also reduces the energy consumption in a manner desirable for the IoT nodes.

In this section, we have analyzed the relative reduction in computational effort for performing cryptographic operations for different modulus sizes from the perspective of IoT nodes. In the following section, we analyze the cost from the perspective of a time-and-budget limited adversary to break the crypto systems of various key sizes.

IV. Breaking Crypto Systems Under the Time-and-Budget-Limited Adversary Model

We present our estimates for the cost of breaking crypto systems under the time-and-budget limited adversary model for IoT in this section.

Copacobana, which is built using FPGA in the year $Y_0 = 2006$ at a Dollar cost of $M_0 = \$10000$, can recover $L_0 = 56$-bit DES symmetric/secret keys in $D_0 = 8.7$ days ([5], [7], [8], [9]). Therefore, the cost of this attack,

$$C_0 = M_0 \times D_0 = 87000\ Dollar\ days. \tag{1}$$

Moore's law dictates that the cost of any successful attack drops by a factor of two, every $\lambda = 1.5$ years. Therefore, applying Moore's law to C_0 from (1), we can estimate the cost of breaking DES in a desired future year Y ([3], [4], [5], [6]) as follows:

$$cost(Y, \lambda) = \left\lceil \frac{C_0}{2^{\left(\frac{Y-Y_0}{\lambda}\right)}} \right\rceil\ Dollar\ days. \tag{2}$$

For example, in the year, $Y = 2014$, the cost of recovering a $L_0 = 56$-bit DES symmetric/secret key is,

$$cost(2014, 1.5) = \left\lceil \frac{87000}{2^{\left(\frac{2104-2006}{1.5}\right)}} \right\rceil = 2158\ Dollar\ days. \tag{3}$$

Equation (3) implies that an adversary with a budget of \$1000, for example, in the year $Y = 2014$ can recover a $L_0 = 56$-bit symmetric key in $D = 2.158$ days.

More generally, the cost of breaking a symmetric cipher that uses a L-bit key, where $L > L_0$, can be expressed as follows:

$$cost^*(Y, L, \lambda) = cost(Y, \lambda) \times 2^{L-L_0} \; Dollar \; days.^1 \quad (4)$$

Using (4), we can determine that an adversary with a budget of \$1000, for example, in the year $Y = 2014$ can recover a $L = 60$-bit key in $D = 34.5$ days.

Equation (4) expresses the cost of breaking a *symmetric key crypto system* such as DES, AES, etc. We now consider the cost of breaking an *asymmetric/public key crypto system* such as RSA, DHKE, etc.

Ecrypt2 [5] expresses the equivalence between symmetric/secret, and asymmetric/public key sizes as follows. Let n denote the number of bits necessary to represent the public key modulus value. Then, the n-bit public key modulus value provides security equivalent to a symmetric key of size, $s(n)$ bits, where,

$$s(n) = \left(\frac{64}{9}\right)^{\frac{1}{3}} \log_2(e) \left(n \ln(2)\right)^{\frac{1}{3}} \left(\ln\left(n \ln(2)\right)\right)^{\frac{2}{3}} - 14. \quad (5)$$

Equation (5) implies that a public key modulus value of length, $n = 512$ bits, for example, is equivalent to a symmetric key of size $s(n) \approx 50$ bits; similarly, a public key modulus value of length, $n = 712$ bits is equivalent to a symmetric key of size $s(n) \approx 60$ bits, and so on.

Combining (4) and (5), we calculate the cost of breaking a public key crypto system in $D = 3$ days, for example, when using a modulus value of size equal to n bits. Specifically, we compute the Dollar cost, $M = cost^*(Y, s(n), \lambda)/D$, for $Y \in \{2014, 2020\}$, $\lambda = 1.5$ years, $D = 3$ days, and for values of $n \in [625, 900]$. We have shown these results in Fig. 2.

Protecting Information in the Smart Grid: Now, as a concrete example, consider the information exchange between a *smart meter, and the utility company server* from Section 1. Assume that the adversary has a budget of $M = \$50000$, and that the house owner is away for 5 days. Further, assume that the smart meter and the utility server secure their communication through DHKE using a modulus of length close to $n \approx 750$ bits in the year $Y = 2014$ (similarly, $n \approx 850$ bits in the year $Y = 2020$), for example. If the adversary knows that he/she can gain far more than $M = \$50000$ through stealing of valuables from the house, then the adversary might consider breaking the crypto system over $D = 3$ days, and then break in to the house on the 4th day, if he/she finds that there is indeed reduction in the power usage. This is a reasonable attack from the attacker's perspective, since in this example:

$$cost^*\left(2014, s(750), \lambda\right)/D < M, \text{ and}$$
$$cost^*\left(2020, s(850), \lambda\right)/D < M.^2$$

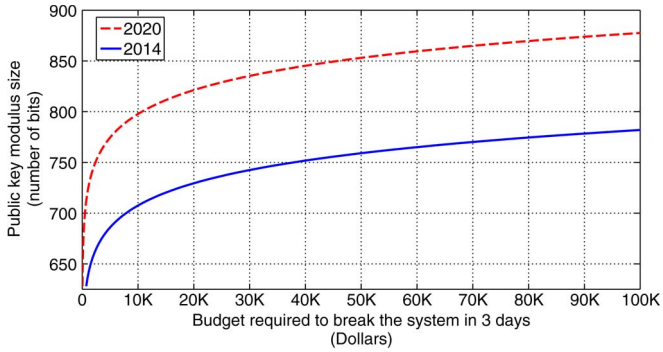

Fig. 2. Public key modulus size versus the Dollar budget required to break the crypto system in 3 days in the year 2014 and year 2020.

However, if the smart meter and utility server secure their communication using a slightly longer DHKE modulus of length $n \approx 800$ bits in the year $Y = 2014$ (similarly, $n \approx 900$ bits in the year $Y = 2020$), then it is infeasible for the adversary to break the DHKE crypto system with the available budget M, since:

$$cost^*\left(2014, s(800), \lambda\right)/D \gg M, \text{ and}$$
$$cost^*\left(2020, s(900), \lambda\right)/D \gg M.$$

Thus, based on the above estimates, the smart meter and utility company server may choose a DHKE modulus of length $n \approx 800$ bits in the year $Y = 2014$ (similarly, $n \approx 900$ bits in the year $Y = 2020$) to remain secure against an adversary with the available budget of $M = \$50000$.

More than Moore, and Less than Moore Growth Rate: We have thus far limited our discussion assuming that the predictions of Moore's law remain accurate; i.e., the computing power doubles every $\lambda = 1.5$ years. However, there are concerns whether Moore's law will continue to hold in the future due to physical limitations in increasing the number of gates per square inch. Thus, it is possible that the technology growth rate may fall under the category of "*Less than Moore*". On the other hand, it is also not unlikely that some disruptive new technology may be developed in the future that would accelerate the growth of technology "*More than Moore*", i.e., beyond the predictions of Moore's law.

Therefore, we consider both the "*Less than Moore*" and "*More than Moore*" cases in our work. For the "*Less than Moore*" case, we choose $\lambda = 3.0$ years; i.e., the computing power doubles every 3.0 years. Similarly, to study the impact

[1]On breaking multiple keys with the same key breaker: Consider the expression, $D = cost^*(Y, L, \lambda)/M$, which characterizes a key breaker capable of recovering a L-bit key at a cost of M Dollars in D days. Since $cost^*(Y, L, \lambda) = cost^*(Y, (L-l), \lambda) \times 2^l$, where $0 \leq l \leq (L - L_0)$, the same key breaker has the capacity to recover 2^l keys, each of length $(L-l)$-bits, in a total of D days. Note that each of the $(L-l)$-bit keys can be recovered in $(D/2^l)$ days. However, these $(L-l)$-bit keys can be recovered only *in sequence* (i.e., not in parallel) with the single key-breaker that costs M Dollars. Let i denote the key number, where $i \in \{1, 2, 3, \ldots, 2^l\}$. Then, the adversary can expect to obtain key number i at the end of $(i \times (D/2^l))$ days.

[2]Note that after breaking a public key modulus (of length $n \approx 850$ bits in the year $Y = 2020$) in D days, the adversary can carry out a second attack for a different house in the next D days, without spending any additional Dollar budget. However, note that if a slightly longer modulus (of length $n \approx 900$ bits in the year $Y = 2020$) is used, the key breaker will be no longer useful to the adversary to carry out any attack within D days!

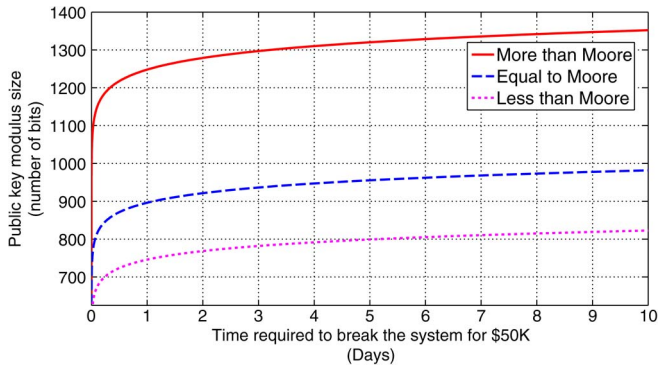

Fig. 3. Public key modulus size versus the number of days required to break the crypto system for $50 K in the year 2025 under different technology growth rates.

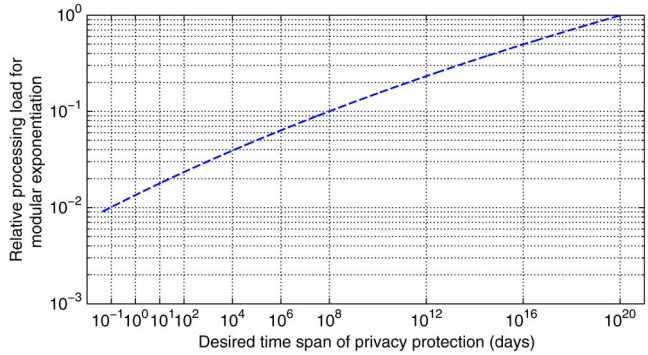

Fig. 4. Relative processing load for modular exponentiation operation versus the desired time span of privacy protection in days.

of the "*More than Moore*" case, we choose $\lambda = 0.75$ years; i.e., the computing power doubles every 9 months.

We have shown the results for different technology growth rates in Fig. 3. Specifically, we compute the number of days, $D = cost^*(Y, s(n), \lambda)/M$, for $Y = 2025$, $n \in [625, 1400]$, $\lambda \in \{0.75, 1.5, 3.0\}$, $M = \$50000$.

Notice from Fig. 3 that when the technology grows steadily at the rate of Moore's law, an adversary with $M = \$50000$ may break a DHKE modulus value of length $n \approx 950$ bits in $D = 5$ days in the year $Y = 2025$. Stated alternatively, a DHKE modulus value of length $n = (950 + s)$ bits is sufficient to remain secure from such a time-and-budget limited adversary, where s denotes the *safety margin*; $s = 50$ bits, for example. However, if the technology progresses at "*Less than Moore*" rate, a smaller modulus value of $n(800 + s)$ bits may be sufficient for securing the communication. Similarly, if the technology progresses at "More than Moore"rate, a larger modulus value of $n = (1320 + s)$ bits may be necessary to remain secure against the above time-and-budget limited adversary.

Processing Load of IoT node versus Desired Time Span of Privacy Protection:

We present the trade-off between the processing load at an IoT node for modular exponentiation versus the desired time span of privacy protection in Fig. 4. We assume that the adversary has a budget of $M = \$50000$.

The relative processing load, $(n^3/3248^3) = 1.0 = 100\%$ when a $n = 3248$-bit modulus is used. Notice from Fig. 4 that such a large modulus value provides security for a very long period of time $\approx 10^{20}$ days. An IoT node can use much smaller modulus to reduce the processing load, as we have shown in Fig. 1. To achieve privacy protection for 10 days, for example, a relative processing load of $<2.0\%$ is sufficient. Even if the desired time span for privacy protection is as large as 10^8 days, the relative processing load is still only $\approx 10\%$. In other words, the processing load at IoT increases only moderately even with very rapid increase in the desired time span for privacy protection.

Our examples above illustrate the relative ease, in terms of processing load, with which an IoT node can achieve sufficient security against time-and-budget limited adversaries. Thus, if the lengths of public key moduli are selected depending on the capabilities of the time-and-budget limited adversary model appropriate for a given application, we can drastically reduce the processing load for IoT nodes.

V. Conclusion

We showed that a small 1024-bit public key modulus, for example, requires an IoT node to perform only 3.1% of the computations relative to a typical 3248-bit modulus. We estimated the number of days, and the budget, in terms of Dollar cost, required for an adversary to break various small-sized keys for the communication between *household smart meters and utility company servers*. Under the time-and-budget limited adversary model, we also considered how to choose the size of modulus values under different technology growth rates, characterized as *More than Moore*, *Moore*, and *Less than Moore*. Finally, we also showed that a relative processing load of $< 2.0\%$ is sufficient to achieve privacy protection for 10 days against an adversary with $50,000 budget.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[2] C. P. Mayer, "Security and privacy challenges in the Internet of things," in *Proc. Electron. Commun. EASST*, 2009, vol. 17, pp. 1–12.
[3] A. K. Lenstra, Key Lengths, Contribution to The Handbook of Information Security, 2004.
[4] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," in *Proc. PKC*, vol. 1751, *Lecture Notes in Computer Science*, 2000, pp. 446–465.
[5] ECRYPT2: Yearly Report on Algorithms and Keysizes, Sep. 2012.
[6] T. Kleinjung, A. K. Lenstra, D. Page, and N. P. Smart, "Using the cloud to determine key strengths," in *Proc. INDOCRYPT*, vol. 7668, *Lecture Notes in Computer Science*, 2012, pp. 17–39.
[7] S. Kumar *et al.*, "How to break DES for Euro 8,980," in *Proc. SHARCS Workshop*, Germany, Apr. 2006, pp. 1–19.
[8] T. Güneysu, T. Kasper, M. Novotny, C. Paar, and A. Rupp, "Cryptanalysis with COPACOBANA," *IEEE Trans. Comput.*, vol. 57, no. 11, pp. 1498–1513, Nov. 2008.
[9] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler, "Breaking ciphers with COPACOBANA -a cost-optimized parallel code breaker," in *Proc. CHES LNCS*, 2006, vol. 4249, pp. 101–118.
[10] A. J. Shipley, Security in the Internet of Things: Lessons from the Past for the Connected Future. [Online]. Available: http://www.newelectronics.co.uk/image-store/articles/58974%5CSecurity-in-the-Internet-of-things.pdf
[11] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[12] [Online]. Available: http://wiki.openssl.org/index.php/Diffie_Hellman
[13] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 1990.