Hindawi
Journal of Computer Networks and Communications
Volume 2019, Article ID 5910528, 10 pages
https://doi.org/10.1155/2019/5910528

Research Article

Application of Spritz Encryption in Smart Meters to Protect Consumer Data

Lincoln Kamau Kiarie,[1] **Philip Kibet Langat**,[1] **and Christopher Maina Muriithi**[2]

[1]*Telecommunication and Information Engineering Department, Jomo Kenyatta University of Agriculture and Technology, P.O. Box 62000–00200, Nairobi, Kenya*
[2]*Electrical and Power Engineering Department, Murang'a University of Technology, P.O. Box 75–102000, Murang'a, Kenya*

Correspondence should be addressed to Lincoln Kamau Kiarie; kamaulincoln@jkuat.ac.ke

Received 12 October 2018; Revised 7 January 2019; Accepted 3 February 2019; Published 26 March 2019

Guest Editor: Amir Rastegarnia

The ongoing upgrade of the electrical power system into a more powerful system known as Smart Grid has both benefits and costs. Smart Grid relies on advanced communication and hence offers better services through improved monitoring, planning, and control. However, enhanced communications make Smart Grid more susceptible to privacy leaks and cyber attacks. Small meters collect detailed consumer data, such as power consumption, which can then become a major source of privacy leakage. Encryption can help protect consumer data, but great care is needed. The popular RC4 (Rivest Cipher 4) encryption has been implemented in the widely deployed smart meter standard—Open Smart Grid Protocol (OSGP)—but has been shown to have major weaknesses. This paper proposes the use of Spritz encryption. Spritz is an RC4-like algorithm designed to repair weak design decisions in RC4 to improve security. A test on performing one encryption took only 0.85 milliseconds, showing that it is fast enough not to affect the operations of a smart meter. Its ability to withstand brute force attacks on small keys is also significantly greater than RC4's ability.

1. Introduction

A key foundation of technological progress is electrical power. The traditional electrical power grid, which handles power from when it is generated until it reaches the customer, has remained relatively unchanged for many years. Demand for power has grown, and the needs have become more complex. To meet these rising challenges, the traditional grid is being upgraded to a better system known as Smart Grid. Smart Grid integrates modern telecommunication to run operations more effectively. It results in superior monitoring with less manpower to collect data, automated fault detection and correction, enhanced power delivery planning, and many other benefits [1–4].

Smart Grid is able to better meet the needs of both the supplier and the consumer. There are less power outages, lower transmission losses, fewer undetected faults, and decreased green house gas emissions. It also allows for distributing power sources, easier integration of renewable energy sources, and more customer choices and can even increase the capacity of the existing electric power networks [5, 6]. A customer who has installed solar panels could at times produce more power than they need. They can then sell this to a utility company, creating a more symbiotic relationship between the two. This cannot happen without effective two-way coordination.

A crucial enabler for Smart Grid is the smart meter. A smart meter provides the utility company more information on electrical consumption than a regular energy meter [4]. It allows for two-way communication with benefits to both the utility and customers. Utilities collect more data to help in planning while incurring less operational costs to do so. Customers can track their usage better, and they can resell energy they generate and have more ways to participate. For these reasons, the European Union (EU) aims to improve

cost-effectiveness by replacing at least 80% of electricity meters with smart meters by 2020 [7].

With all these promises that Smart Grid offers, there is a major concern that must be addressed. This concern is familiar to anyone working on an interconnected system, whether smartphones or computers connected to the Internet. This is the threat of cyber security and privacy leakage. Smart Grid consists of plenty of control and monitoring information being transmitted, and it serves millions of users. Securing these data is thus extremely crucial, especially given the recent rise of cyber security threats.

Hackers can manipulate signals maliciously or access and monitor information that is private. Manipulating *control* information can cause the system to respond in unexpected ways, ranging from power outages to damage of equipment [8]. Energy theft can be done by altering billing information [9] and a Denial-of-Service (DoS) attack can disrupt power [10]. Hackers accessing *monitoring* information can misuse it in a number of ways: a robber can tell which electrical devices people use [11], whether they are home [12], how many people are there [13], and even when they eat or bath [14]. Under the right conditions, they can even view an image of what is being watched on television [15]. User profiling and household classification (e.g., based on income level) are possible [16]. Such leakage of personal information can have severe, far-reaching consequences, ranging from well-planned robberies and spying to aggressive marketing and large-scale surveillance. Health insurers can also charge customers different rates based on their exercise activities (or lack thereof) [17].

These concerns have not gone unnoticed by the public, and there are lobby groups strongly campaigning against the user of smart meters [18]. Privacy is one of their major concerns. If the concerns of customers are ignored, successful adoption of smart meters will be slowed down. Given the potential benefits offered by smart meters, this would be a regrettable loss in this digital era. This paper seeks to help reduce the privacy problem by using an encryption technique known as Spritz. Before explaining how it works, we first look at how smart meters cause such invasive privacy problems.

Privacy leakage starts off when details of a customer's electricity usage can be identified. A customer's private activities can be revealed if information was leaked about the electrical devices they use, what time of the day they use them, and how long they were used. For such a leakage to occur, it might be thought that spying devices would have to be connected to the sockets, but it is not so. This information can in fact be extracted from the overall power consumption patterns. Using a technique called nonintrusive load monitoring (NILM) [19], it is possible to identify specific electrical devices, based on how they consume power. Each device has a unique pattern (known as a *load signature*)—based on its power rating and mode of operation—that allows an observer to identify it from a graph of the total power consumed. This technique has been refined over time to make it very powerful [11]. Figure 1 illustrates how this can be done using data from typical household [20].

An attacker can obtain the overall consumption information by wiretapping [21] or traffic analysis [22]. They can then analyse the data at their convenience. Eavesdropping is even easier with wireless transmission, and thus, proper security measures are needed.

To solve the security problem of Smart Grid, one could try borrowing standard methods (such as encryption) used in computer security, but this would fail for two reasons. First, Smart Grid devices use light-weight embedded systems (e.g., on smart meters), which have limited processing power and memory capacity. These devices may be unable to handle the computational load needed in implementing some forms of encryption. Second, some sections of Smart Grid, such as fault detection, have very small time allowances (i.e., latency). A delay in transmitting an urgent message with information on protection could be as bad as blocking the message all together, and this can cause system failure. As an example, power substation networks based on IEC 61850 use a GOOSE (Generic Object-Oriented Substation Event) communication module. This transmits protection messages and is time-critical, having a maximum time allowance of 3 ms [23]. A balance between security and performance is needed.

A major effort to mitigate these problems was by the Open Smart Grid Protocol (OSGP) [24]. This had been deployed in millions of smart meters installed worldwide but was found to have serious security weaknesses [25, 26]. The first version of OSGP used the popular stream cipher RC4 (Rivest Cipher 4) which has been applied to various protocols such as Transport Layer Security (TLS). An efficient encryption system is thus needed that would overcome the challenges of RC4 while maintaining its strengths. We now turn to an encryption that has the benefits of RC4 without most of its problems.

1.1. Spritz Encryption. Spritz is a relatively new encryption algorithm that was designed as a "drop-in replacement" for RC4 encryption with the aim of eliminating the major weaknesses in RC4 [27]. This paper examines its suitability in smart meters. Spritz was designed to have improved security by revisiting particular design decisions and improving on them, in light of known attacks. Spritz, like RC4, is a stream cipher that typically works byte by byte. It consists of 6 other registers in addition to a state vector S; thus, the number of possible states it has is at most $N^6N!$ States. Using the standard value $N = 256$, this computes to $\approx 2.415 \times 10^{512}$ states that it is very difficult to carry out successful cryptanalysis against it. The best attack published as of this writing for recovering the state of Spritz requires 2^{1247} steps $\approx 2.423 \times 10^{375}$ [28]. Although this is an improvement from brute force, it is still well outside the reasonable range of our best supercomputers.

A number of factors that make Spritz superior to RC4 include

(i) Existence of weak keys in RC4 [29].

(ii) Biases in the output stream that is generated [30, 31].

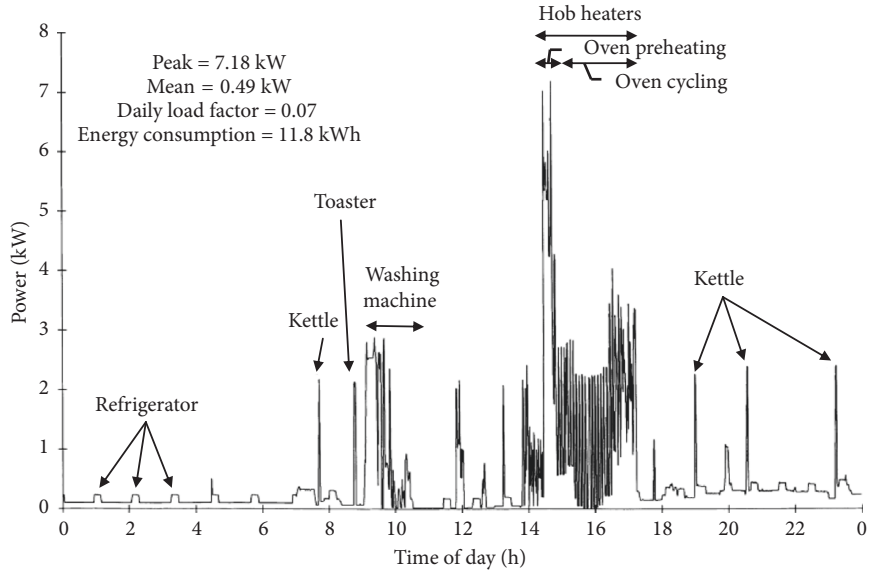

Figure 1: Household profile with devices identified from recordings on a one-minute time base over a 24-hour period.

(iii) A lower state recovery attack (complexity of 2^{241} steps compared to 2^{1247} steps in Spritz [28, 32]).

(iv) RC4 has key collisions [33]. These occur when two different keys produce the same state. This results in colliding keys producing the same output stream.

(v) Spritz also provides additional cryptographic capabilities which can provide more services in addition to security (e.g., hashing, message authentication code, and authenticated encryption).

In order to compare Spritz and RC4 on fair ground, the additional capabilities in the last point have not been utilized in Spritz and only encryption is compared between the two. These extra services would however find beneficial application in smart meters, such as detecting manipulation of electricity consumption data.

1.2. Related Works. Due to the far-reaching effects of privacy violation, it is of little wonder that this topic has attracted the interest of many researchers from varying fields. Solutions to this problem can be broadly classified into two. The first group is legal-/policy-based which involves creating and implementing policies to regulate how customer information is disclosed to third parties [34]. The second group is technical based, which uses algorithms to make it harder for an attacker to obtain the data, whether or not they are interested in keeping the law. A leading security expert [35] explained, "it is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics." Both legal and technical solutions are needed. This work addresses the problem from a technical approach.

One way of solving privacy leakage has been proposed by Kalogridis et al. [36] and is known as battery-based load hiding (BLH). Power supplied to a house is first fed to a battery before being used; then, an algorithm flattens the load profile to an almost constant value. This prevents data mining through NILM. BLH has been improved on by Yang et al. [37] by mathematically maximizing the error between the load demanded by a home and the external load seen by a smart meter. BLH in general has the shortcoming of introducing power losses since no battery is 100% efficient.

Another approach is through the use of a photovoltaic converter [16]. The technique modifies maximum power point tracking (MPPT) in order to generate fake load signatures. Though it does not introduce the power losses in by using intermediary batteries, it requires users to install solar. The design also reduces solar power's efficiency to attain privacy.

Rial and Danezis [14] propose a scheme where a user combines meter readings with a certified tariff policy to produce a final bill. The bill is then transmitted to the provider alongside a zero-knowledge proof that shows the calculation is correct, without leaking any additional information.

Efthymiou and Kalogridis [38] propose the use of anonymization of smart metering data. They use a trusted third party to ensure that the frequent electrical metering data sent by smart meters are anonymous. The utility is thus able to get the information it needs for its operations, but the high frequency data does not need to be attributed to a particular meter. In their conclusion, they admit that the method "may not offer sufficient smart metering privacy protection [but] contributes an additional layer of security towards that direction" [38].

Homomorphic encryption has also been recommended as a way of protecting privacy. It is a fairly recent development in cryptography [39] that allows one to manipulate encrypted data without decrypting it. Thus, it can keep not only an eavesdropper away but also the utility company from all the customer data. Rather, they will access aggregate usage data from various customers [10]. It does however come with significant computational cost. Smart meters are generally light-weight devices, and some may not handle

homomorphic encryption. The smart meters also need to have a trusted component and enjoy a certain level of autonomy [40].

End-to-end encryption involves encrypting the data between a customer and a utility. Both parties have a shared key enabling them to encrypt and decrypt the data. A strong and well-implemented encryption algorithm should thwart the efforts of the eavesdropper to listen in on what is being transmitted. In the context of smart meters, computational complexity and security need to be balanced to avoid either having a system that is very fast but insecure or one that is extremely slow but secure.

Table 1 below compares the above methods on a number of factors: do they introduce power loss? Is there a significant increase in the computational or communication overhead? Do you need to have a trusted third party for them to work? Can the utility company access the customer's data? This comparison is helpful in choosing a method suitable to a particular setting with its own unique threats. It could further help in combining options and having multiple layers of security to curb a range of threats.

2. Materials and Methods

Spritz encryption is tested and compared to RC4 to determine if it provides better security. Electrical consumption data of a typical household over 24 hours of minute-by-minute observation were used. After performing encryption, encrypted data were plotted for both algorithms to demonstrate the transformation on the data. A random plot was expected for both.

Generation of encryption keys is a crucial step in cryptography since a predictable way of doing so would compromise security, no matter how strong encryption is. We thus used a scheme that relies on hash functions to generate a key. Hash functions are algorithms which take an input of variable size and produce an output with a fixed number of bits (e.g., an input of a 2 MB image can produce a specific output of a 256 bit hash value). They are thus a valid choice for generating keys for use in encryption [41]. Among the existing hash functions available, the strongest and most reliable hash functions belong to a family of algorithms developed by the National Security Agency (NSA) known as Secure Hash Algorithm (SHA) [42]. Of this, the one producing the largest output is SHA512, with an output of 512 bits. We used the meter number of a smart meter and computed its SHA512 value. From this value, we used the first 128 bits which were then used for encryption.

In order to examine the effectiveness of the aforementioned key generation method, a comparison was done between keys generated by consecutive meter numbers. Keys produced from three adjacent meter numbers were compared to see if they had an obvious relationship. A significant similarity between the bits of neighbouring keys would imply that a hacker has more options in trying to crack your data. A compromised meter would make the meters around it also insecure, and the effect could be scaled upwards.

This approach of key generation was subjected to an additional test. During key generation, some patterns could exist that would help an attacker obtain the key. Assuming that meter numbers are consecutive, would nearby keys have an obvious relationship? A meter number was compared to the next 100 meter numbers after it to see if the bits appeared similar. A strong correlation is an indicator that the key generation method is weak and could be exploited by a hacker.

The speeds for RC4 and Spritz were compared to see how their performances relate. Spritz is a more complex algorithm than RC4 and was thus expected to take longer. However, given that RC4 has been deployed in smart meters (under the OSGP standard), what would be needed is to see how much slower Spritz would be. If the time taken is of the comparable (e.g., the same order of magnitude), then it would be a suitable candidate for smart meter encryption.

To compare RC4 and Spritz in terms of strength, brute force was used to retrieve the key from the encrypted data of both ciphers. Short keys were used to make brute force attack practical since this entails searching through all possible keys. The time taken for each was computed, and a plot was made to compare which of the two is harder to break.

For implementing the above, a code was written and run on Octave (an open-source software very similar to Matlab). All experiments were done on an Intel(R) Core(TM) i5-2540M CPU @ 2.60 GHz.

The details of our method are expounded below.

2.1. Data Used. The data shown in Figure 1 were used to test encryption. They were conveniently chosen because they already highlight how devices in use can be deduced by analyzing overall power consumption (i.e., using non-intrusive load monitoring). The source article did not provide the original data points; thus, image processing was used to retrieve them. The steps used were as follows:

(1) An image of the graph with the data was read into an array.

(2) A suitable thresholding value was used to convert the image in the matrix from grayscale to a black and white binary image. With this, points on the plot which are part of a line/text will be black (represented by 0), while other parts of the image will be white (represented by 1).

(3) Pixel coordinates of the origin (X_0, Y_0) and of the top left corner of the plot area $(X_{\max}, Y_{\max})$ were located.

(4) For each value of x (i.e., along the horizontal axis), edge detection was used to find where the line is located. This was done by a vertical linear search done pixel by pixel, until the value changed from white to black then back to white. This was recorded as the y value.

(5) The values of the vertical axis were scaled by multiplying the ratio of actual graph values to pixel values.

(6) Scaling along the horizontal axis was done using linear interpolation.

Table 1: Comparison of various privacy enhancing techniques.

Technique	Power loss introduced	Computational overhead introduced to grid	Communication overhead introduced	Trusted third party needed?	Can utility company access data?
Battery load hiding (BLH)	Yes	No	No	No	No
Solar PV convertor	Yes	No	No	No	No
Anonymization	No	Yes	Yes	Yes	No
Homomorphic encryption	No	Yes	Yes	Yes	No
End-to-end encryption	No	Yes	Yes	No	Yes

These data points were then plotted to see how well they were, compared to the original graph. As seen from Figure 2, the shape of the plot was adequately captured. This was sufficient for testing encryption.

2.2. Key Selection and Analysis. Smart meter security needs to consider the fact that a large number of users (customers) will need to implement any new solution being proposed. Since an average size country would have millions of electricity users, scalability is very important. A large number of keys need to be generated. It would be convenient to have the keys correspond to the meter numbers to ease the process of generating keys in bulk. However, the keys generated from consecutive meter numbers should be sufficiently different, failure to which an attacker could find a way to exploit such a relationship.

It is important to note that the method used to generate the key *must* be kept a secret. Just as you would not tell others how you choose your passwords or PIN numbers, the same is true with keys for smart meters. Anyone wishing to implement this work would need to modify this method to generate their keys in a manner that is not publicly known.

Our key generation approach used the following steps:

(1) The meter number was express as a string of characters

(2) The hash value of the meter number was computed using SHA512

(3) The first 128 bits of the hash value obtained were chosen as the key (the one produced by the original meter number will henceforth be called *Key1*)

To examine the security of the above technique, keys produced by adjacent meter numbers were compared to see if the bits are similar. Using the meter number that follows the one above, a key designated $Key1^+$ was produced. And, using the meter number just before the original one, a key designated $Key1^-$ was produced.

If a large percentage of bits are identical between *Key1* and $Key1^+$ (or $Key1^-$), it would mean that the technique is susceptible to attack from an attacker able to access a neighbouring meter. The *hamming distance* is a helpful quantity in this regard. It is defined as the number of bits in which two binary quantities disagree [43].

A second test on the key generation was done by finding the hamming distance between the key from meter 1 (i.e., *Key1*) and the key produced by the next 100 meters. A plot of these values was produced, and the average hamming distance was computed. If the key generation method is good, this average should be close to 50%.

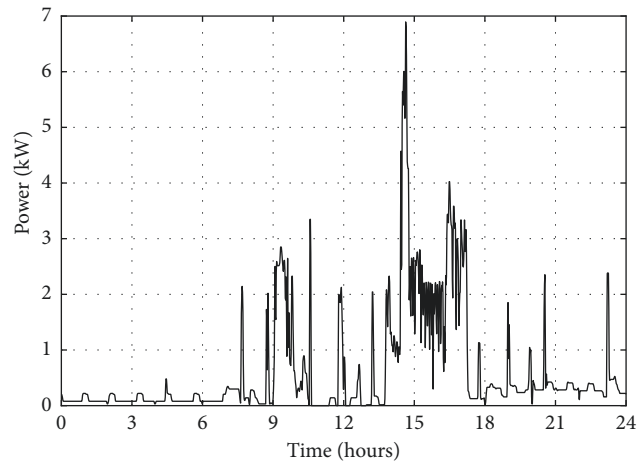

Figure 2: Approximation of typical household energy usage.

2.3. RC4 Encryption. The steps followed were

(1) RC4, being a stream cipher, performs encryption one byte (i.e., 8 bits) at a time. Its input thus needs to be in byte form. Each data point was converted by splitting it into a pair of bytes (i.e., a 16 bit quantity allowing a range of 0–65535 W, which was sufficient for the data used). Big-endian format was used, in which the most significant byte is stored towards the beginning of an array.

(2) A key stream (which is a sequence of pseudorandom bytes) was generated using RC4 algorithm. The algorithm uses a key for the generation. *Key1* was used in this step.

(3) XOR operation was applied between each pair of bytes from step 1 and each successive pair of bytes from step 2. This operation results in a pair of bytes serving as the ciphertext (i.e., encrypted form of the data).

(4) The encrypted pairs of bytes were then merged to form a value that can be transmitted. This essentially reverses the splitting of step 1. These data were plotted, and the resulting graph is shown in Figure 3.

These steps are summarized in the block diagram of Figure 4.

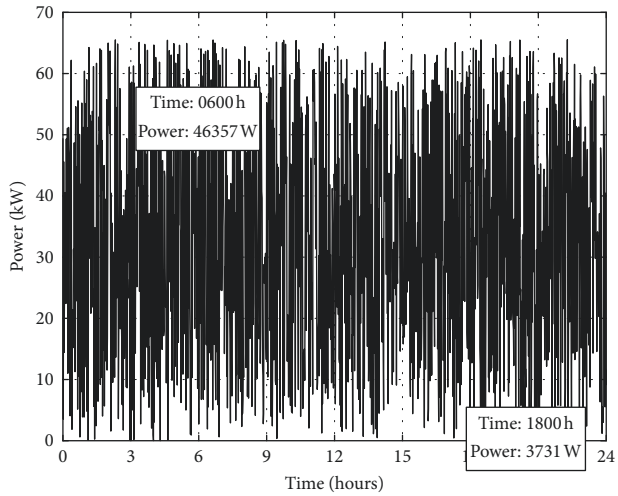

Figure 3: Encryption with RC4 using *Key1*.

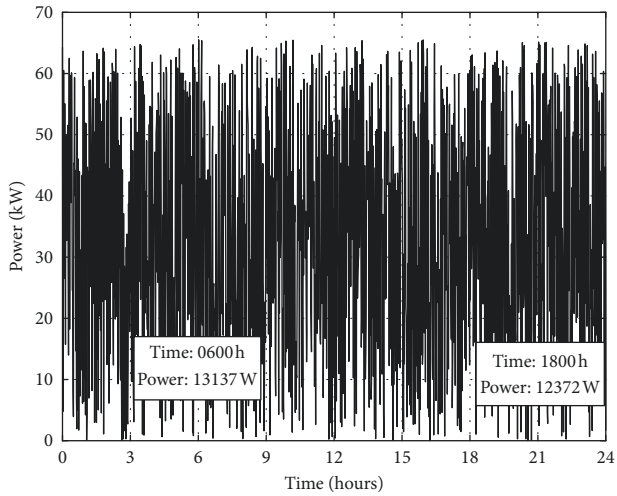

Figure 5: Encryption with Spritz using *Key1*.

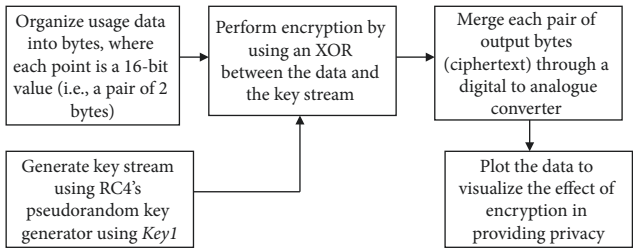

Figure 4: Applying RC4 to electrical usage data.

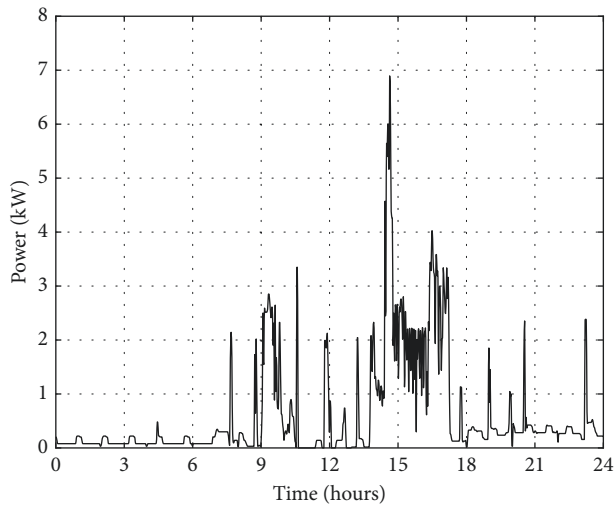

Figure 6: Plot of values after Spritz decryption using *Key1*.

2.4. Spritz Encryption. A similar procedure to the one just described for RC4 was used:

(1) Spritz is also a stream cipher, and it encrypts data byte by byte. Each data point was thus converted by splitting it into a pair of bytes.

(2) A key was fed into the "*keySetup*" function for Spritz, and then, encryption was done (the details for how Spritz encryption works are available in [27]).

(3) Encrypted pairs of bytes were then merged to form a value that can be transmitted. This essentially reverses the splitting of step 1. These data are plotted, and the resulting graph is shown in Figure 5.

(4) These encrypted data are then converted back to byte pairs and decrypted using the same key. This is to confirm that the data received at the utility is the same as the original once it is decrypted. The plot of decrypted data is shown in Figure 6.

Figure 7 summarizes the overall process.

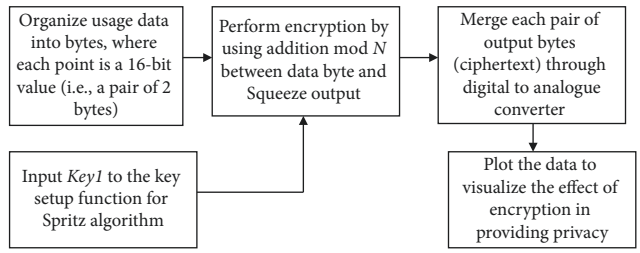

Figure 7: Applying Spritz to electrical usage data.

2.5. Attacks on Attenuated Ciphers. An attack that can work across different ciphers is brute force which is also known as exhaustive key search. In this work, keys of 128 bits were proposed for the implementation of smart meters, giving a total of 2^{128} ($\cong 3.403 \times 10^{38}$) possible options. Even if it were possible to test 1 billion keys every second, it would take 10^{22} years to search through all keys and thus break the cipher.

In light of the above computational barrier, it would be practically impossible to compare the strength of the two ciphers when both use 128 bit keys. Instead, short keys were applied to both RC4 and Spritz, and the time taken to recover the key by brute force was computed. This can be

compared to testing the effect of a disease-causing agent (e.g., bacteria) on an animal by using its attenuated (weakened) version. The results can then be extrapolated from there. To compare the strength of RC4 and Spritz, smaller keys were applied to the original data. The time taken for each to be broken was then plotted. This plot would indicate which of the two ciphers is more secure from a brute force attack.

Keys were chosen, ranging from 1 to 8 bits in length. They were then used to encrypt the household data using RC4 and then using Spritz. The time taken for each case was computed. A graph of the time taken to retrieve the key was plotted against the length of the key (in bits).

3. Results and Discussion

From the graphs of the encrypted data, information privacy can be observed. Without the key, a person snooping on the usage data during transmission would be unable to deduce details of the electrical device in usage. NILM or any other data mining technique would not reveal user behaviour. Encryption thus conceals power usage patterns.

Both RC4 and Spritz offer a degree of privacy. RC4 encryption is shown in Figure 3, while Spritz is shown in Figure 5.

Two data points were labelled in each graph, corresponding to 6 am and 6 pm, to help distinguish the random figures obtained from the two procedures. As Table 2 shows, the two random-looking graphs encrypt the data differently.

Decryption for Spritz using the same *Key1* restores the original data as shown in Figure 6 (when RC4 was decrypted, an identical plot was obtained, and thus, it has not been included here).

3.1. Key Selection and Analysis. The key was generated using a subset of the hash function for the meter number. An actual meter number was used. The first and last 3 digits of this meter number were 221...006 (middle digits are removed since they are from an actual, 11-digit electrical meter number). The result obtained by taking the first 128 bits of its hash function (using SHA512) was found to be (in hexadecimal)

$$Key1 = 0\text{xca5c1d000455c154023829d96301c2fc}$$

To check if this key is closely related to the one generated by the adjacent meter numbers, a comparison was made with two other keys. Taking the meter number just after the one used (i.e., 221...007) and the one just before (i.e., 221...005), two keys were obtained using the same technique. These were denoted as $Key1^+$ and $Key1^-$, respectively. Their values were found to be

$$Key1^+ = 0\text{xe1f23f98c4edc8e8b1b6ad62f1a49c19}$$
$$Key1^- = 0\text{x58cb4141b4d158f903dec6562e962458}$$

With the above, the hamming distance, $d(\mathbf{v}_1, \mathbf{v}_2)$, was computed to see if there is a likely relationship between keys from adjacent meters:

Table 2: Test data points for RC4 and Spritz.

Algorithm	Value at 6 am (kW)	Value at 6 pm (kW)
RC4	46.357	3.731
Spritz	13.137	12.372

$$d(Key1, Key1^+) = 61$$
$$d(Key1, Key1^-) = 63$$
$$d(Key1^+, Key1^-) = 62$$

This indicates that the difference between the bits of *Key1* and $Key1^+$ is ≅47.7%, the difference for *Key1* and $Key1^-$ is ≅49.2%, and that for $Key1^+$ and $Key1^-$ is ≅48.4%. These results indicate that keys generated by adjacent meter numbers are significantly different and are unlikely to provide a hacker with information on the relationship between keys. Compromise of one meter is unlikely to result in compromise of an adjacent one.

An analysis of the next 100 meter numbers and the percentage variation in the keys they produce are shown in Figure 8. The average for these values was 49.39%. This is a good indicator that the key generation method is not easily predictable by an attacker, even if they have access to one of the keys of a particular meter number.

3.2. Time Taken for RC4 and Spritz Encryption. When 10 trials of RC4 encryption were done, the average run time was found to be 0.45 milliseconds. For Spritz, the average for encryption over 10 runs was found to be 0.85 milliseconds. Taking the ratio shows that Spritz is only *1.889* times slower than RC4. Thus, if Spritz can offer superior security over RC4, it would be a superior option since its time performance is less than 2 times slower. Spritz has potential to work well even on a light-weight device like the smart meter.

4. Attacks on Attenuated Ciphers

To compare the ability of RC4 and Spritz in terms of resisting a brute force attack, both ciphers were deployed using keys that were deliberately shortened. Keys were chosen ranging from 1 to 8 bits in length. The value of the keys chosen consisted of a binary string of all 1s (e.g., for 5 bits, 11111 was used). This setup creates a worst-case scenario requiring the longest possible time to break the cipher. The expected duration of a typical case would be half as long since there is a 50% of having a 1 or a 0.

For each key, encryption of the household data was done using RC4 and then using Spritz. An exhaustive search was then done for each algorithm to try and decrypt to retrieve the correct key. The time taken for each case was computed. A graph of the time taken to retrieve the key was plotted against the length of the key (in bits).

As can be seen from Figure 9, the time taken to retrieve the key is lower in RC4 than it is in Spritz. For the case of an 8 bit key, the time needed for RC4 was 52.3 seconds while the time for Spritz was 167.7 seconds. Dividing the two durations results in a ratio of **3.21**.

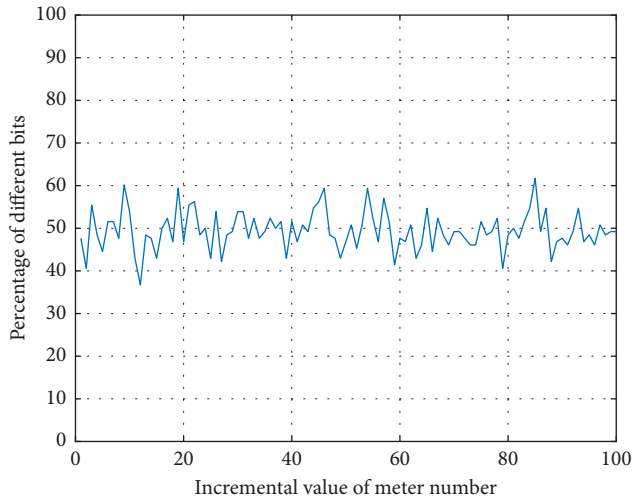

Figure 8: Variation in keys bits as the meter number increases sequentially.

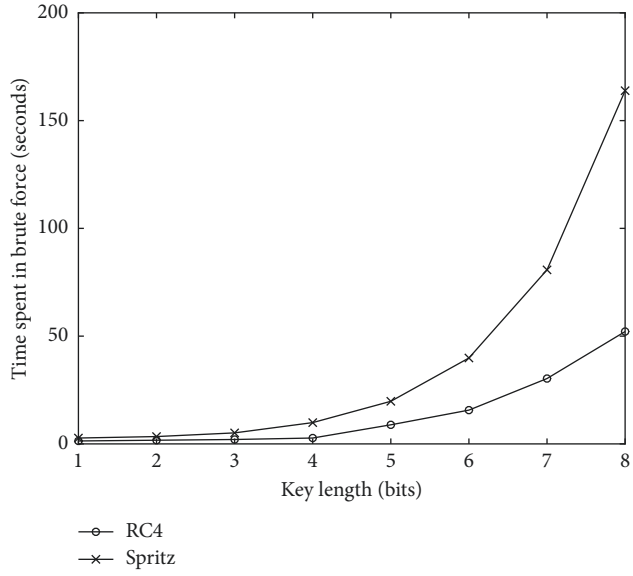

Figure 9: Plot of time taken for brute force for RC4 and Spritz.

This means that an attacker whose goal is to break RC4 has a much easier task than one who is attacking Spritz. This test reveals Spritz to be significantly more resistant to attack than RC4.

As the key length becomes longer, the time needed to break both will increase. However, Spritz will still require a longer time to break. Thus an attacker with resources that barely manage to break RC4 will still not be able to break Spritz.

5. Conclusions

The importance of privacy cannot be disregarded in the deployment of Smart Grid, given our high dependence on electricity and the rising cyber threats. Users are now more sensitive than ever to privacy invasion and thus ignoring their concerns would hinder Smart Grid adoption. Improving privacy would go a long way in building customer confidence in Smart Grid and meeting their needs better. A solution is needed that is both secure and efficient enough to work within the computationally restrictive Smart Grid environment.

This paper recommends the use of Spritz encryption in smart meters, and to the best of the authors' knowledge, this has not been done before. Spritz encryption was found to be an effective way of providing smart meter users with privacy. Due to its speed, its usage would not result in poor system performance due to excessive delays. While its speed is less than 2 times slower than RC4, it would take approximately 3 times longer to break by brute force. Since RC4 has been deployed in operational smart meters (although found to be weak), Spritz is likely to work with better security and without introducing performance issues.

Proposing Spritz as a viable method for smart meters also helps with algorithm agility [25]. History has shown that encryption techniques become weaker with time, as they are subjected to more scrutiny and attackers get faster machines and better algorithms to work with. Having multiple-tested and viable encryption schemes makes replacement easier to implement when a current scheme is broken. It is better to have several tested options and have them ready for deployment than to get into panic the day a researcher publishes a paper, detailing how they broke smart meter encryption.

This work also proposes a reliable method for generating a large number of keys that do not produce related keys. Anyone willing to implement this should use another approach that has similar desirable properties (e.g., recursive use of a hash function and different hash subset for keys).

Further work would involve implementing the additional security features provided by Spritz. These include authentication of the data. Additionally, this algorithm can be implemented on a microchip and installed in a smart meter. The chip would have fewer overheads than simulation but also slower processing. An examination of its performance would provide the final verdict on its effectiveness in smart meters.

Data Availability

The electrical household data used to support the findings of this study are included within the article. The meter number used while generating the encryption key was not given in full since it belongs to an actual electrical meter installed in a customer's premise.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] M. Kezunovic, J. D. McCalley, and T. J. Overbye, "Smart grids and beyond: achieving the full potential of electricity systems," *Proceedings of the IEEE*, vol. 100, pp. 1329–1341, 2012.

[2] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid–the new and improved power grid: a survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2011.

[3] Z. Fan, G. Kalogridis, C. Efthymiou, M. Sooriyabandara, and J. McGeehan, "The new frontier of communications research : smart grid and smart metering," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pp. 115–118, Passau, Germany, April 2010.

[4] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Smart meters for power grid: challenges, issues, advantages and status," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 6, pp. 2736–2742, 2011.

[5] Z. Fan, "Distributed demand response and user adaptation in smart grids," in *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pp. 726–729, Dublin, Ireland, May 2011.

[6] X. Yang, X. Zhang, J. Lin, W. Yu, X. Fu, and W. Zhao, ""Data integrity attacks against the distributed real-time pricing in the smart grid," in *Proceedings of the 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, Las Vegas, NV, USA, December 2016.

[7] D. Koo, Y. Shin, and J. Hur, "Privacy-Preserving aggregation and authentication of multi-source smart meters in a smart grid system," *Applied Sciences*, vol. 7, no. 10, p. 1007, 2017.

[8] K. Tazi, F. Abdi, and M. F. Abbou, "Review on cyber-physical security of the smart grid: attacks and defense mechanisms," in *Proceedings of the 2015 International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 1–6, Marrakech, Morocco, December 2015.

[9] Y. Liu, S. Hu, and T. Y. Ho, "Vulnerability assessment and defense technology for smart home cybersecurity considering pricing cyberattacks," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, vol. 2015, pp. 183–190, Austin, TX, USA, November 2015.

[10] M. Badra and S. Zeadally, "An improved privacy solution for the smart grid," *International Journal of Network Security*, vol. 18, no. 3, pp. 529–537, 2016.

[11] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 356–362, Toronto, ON, Canada, July 2012.

[12] W. Kleiminger, C. Beckel, T. Staake, and S. Santini, "Occupancy detection from electricity consumption data," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pp. 1–8, Roma, Italy, November 2013.

[13] I. Rouf, H. Mustafa, M. Xu, W. Xu, R. Miller, and M. Gruteser, "Neighborhood watch: security and privacy analysis of automatic meter reading systems," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS 2012)*, pp. 462–473, Raleigh, NC, USA, October 2012.

[14] A. Rial and G. Danezis, "Privacy-preserving smart metering," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, pp. 49–60, Chicago, IL, USA, October 2011.

[15] U. Greveler, P. Glösekötterz, B. Justusy, and D. Loehr, "Multimedia content identification through smart meter power usage profiles," in *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, Galway, Ireland, October 2012.

[16] A. Reinhardt, G. Konstantinou, D. Egarter, and D. Christin, ""Worried about privacy? Let your PV converter cover your electricity consumption fingerprints," in *Proceedings of the IEEE International Conference on Smart Grid Communications*, Miami, FL, USA, November 2015.

[17] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM. Workshop on Embedded Sensing Systems for. Energy-Efficiency in Building*, pp. 61–66, Zurich, Switzerland, November 2010.

[18] J. Hart, "Stop smart meters!," 2018, http://stopsmartmeters.org/.

[19] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1871-1872, 1992.

[20] E. L. Quinn, "Privacy and the new energy infrastructure," Tech. Rep., Center for Energy and Environmental Security (CEES), Boulder, CO, USA, 2009.

[21] K. Jain, "Security based on network topology against the wiretapping attack," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 68–71, 2004.

[22] C. V. Wright, S. E. Coull, F. Monrose, and C. Hill, "Traffic Morphing : an efficient defense against statistical traffic analysis," in *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February 2009.

[23] W. Wang and Z. Lu, "Cyber security in the smart grid: survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[24] Protocol: Open smart grid protocol (OSGP), ETSI GS OSG 001 V1.1.1, 2012, http://www.etsi.org/deliver/etsi_gs/OSG/001_099/001/01.01.01_60/gs_osg001v010101p.pdf.

[25] K. Kursawe and C. Peters, "Structural weaknesses in the open smart grid protocol," in *Proceedings of 10th International Conference on Availability, Reliability and Security (ARES 2015)*, pp. 1–10, Toulouse, France, August 2015.

[26] P. Jovanovic and S. Neves, "Practical cryptanalysis of the open smart grid protocol," in *Fast Software Encryption*, pp. 297–316, Springer, Berlin, Heidelberg, 2015.

[27] R. L. Rivest and J. C. N. Schuldt, *Spritz—A Spongy RC4-like Stream Cipher and Hash Function*, IACR Cryptology ePrint Archive, Santa Barbara, CA, USA, 2014.

[28] S. Banik and T. Isobe, "Cryptanalysis of the full Spritz stream cipher," in *Fast Software Encryption*, pp. 63–77, Springer, Berlin, Heidelberg, 2016.

[29] A. Roos, "A class of weak keys in the RC_4 stream cipher," Vironix Software Laboratories, 1995, http://impic.org/papers/WeakKeys-report.pdf.

[30] S. Sen Gupta, S. Maitra, G. Paul, and S. Sarkar, "(Non-) random sequences from (non-)random permutations-analysis of RC_4 stream cipher," *Journal of Cryptology*, vol. 27, no. 1, pp. 67–108, 2014.

[31] R. Bricout, S. Murphy, K. G. Paterson, and T. van der Merwe, "Analysing and exploiting the Mantin biases in RC_4," *Designs, Codes and Cryptography*, vol. 86, no. 4, pp. 743–770, 2018.

[32] A. Maximov and D. Khovratovich, "New state recovery attack on RC_4," in *Proceedings of the Annual International Cryptology Conference 2008*, vol. 5157, pp. 297–316, Santa Barbara, CA, USA, August 2008.

[33] A. Jana and G. Paul, "Revisiting RC_4 key collision: faster search algorithm and new 22-byte colliding key pairs," *Cryptography and Communications*, vol. 10, no. 3, pp. 479–508, 2018.

[34] E. L. Quinn, "Smart metering and privacy: existing laws and competing policies," A Report for the Colorado Public Utilities Commission, 2009.

[35] B. Schneier, *Applied Cryptography: Protocols, Algorithm, and Source Code in C*, John Wiley & Sons, Hoboken, NJ, USA, 1996.

[36] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda, "Privacy for smart meters: towards undetectable appliance load signatures," in *Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications*, pp. 232–237, Gaithersburg, MA, USA, October 2010.

[37] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, and P. McDaniel, "Minimizing private data disclosures in the smart grid," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 415–427, Raleigh, NC, USA, October 2012.

[38] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications*, pp. 238–243, Gaithersburg, MA, USA, October 2010.

[39] C. Gentry, *A Fully Homomorphic Encryption Scheme*, Stanford University, Stanford, CA, USA, 2009.

[40] Z. Fan, P. Kulkarni, S. Gormus et al., "Smart grid communications: overview of research challenges, solutions, and standardization activities," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 21–38, 2013.

[41] W. Stallings, *Cryptography and Network Security Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 5th edition, 2011.

[42] FIPS PUB 180, *Secure Hash Standard*, National Institute of Standards and Technology, Gaithersburg, MA, USA, 2012.

[43] W. Stallings, *Data and Computer Communications*, Prentice-Hall, Upper Saddle River, NJ, USA, 8th edition, 2007.