Flexible Data Authentication Evaluated for The Smart Grid

Kelsey Cairns, Carl Hauser, Thoshitha Gamage
School of Electric Engineering and Computer Science
Washington State University, Pullman, WA 99164-2752
kcairns@wsu.edu, {hauser, tgamage}@eecs.wsu.edu

***Abstract*—This paper explores Time-Valid One-Time-Signature (TV-OTS) as a data authentication protocol for potential use in smart grid applications. TV-OTS is highly configurable with computationally lightweight signing and verification processes, making it a strong candidate for smart grid use. A detailed analysis of security against brute force attacks is presented, which is critical to understanding the parameters under which TV-OTS is reliably secure. This analysis is used to choose applicable parameters for latency tests of a full TV-OTS implementation. Performance results are favorable, with the cost of combined signing and verifying reaching under 10 milliseconds.**

I. Introduction

Electric power delivery is arguably one of the most essential and widely used critical infrastructures, but aging technologies are becoming inadequate for the evolving power grid [1]. The future necessity of a redesigned grid is widely accepted throughout the research community [2]–[4]. The new grid is envisioned to be a smart grid in which situational awareness over large geographic regions will allow greater automation, protection, regulation and operating efficiency. These features rely on sufficient network communication, with messages digitally secured through encryption and authentication. However, most of today's security standards were not created to match the real-time needs of the smart grid, leading to investigation into new protocols. This paper analyzes the TV-OTS data authentication protocol [5] as a solution for multicast authentication in the context of the smart grid.

TV-OTS provides low latency multicast message signing and verification fitting the real-time needs of the grid. As such, TV-OTS is one of the first protocols to simultaneously fulfill three key requirements:

1) Secure multicast
2) Low latency
3) Sufficient lifespan, even at high data rates[1]

To the best of our knowledge, TV-OTS has yet to face rigorous examination. We provide both theoretical and practical analysis intending to showcase TV-OTS in the context of the smart grid and general real-time multicast applications.

The remainder of this paper is organized as follows. An overview of communication in the smart grid is given in Section II. Sections III and IV discuss the benefits of TV-OTS and its basic mechanisms. A theoretical security analysis is presented in Section V which acts as a basis for the experiments described in Section VI. Sections VII, VIII and IX discuss applied security, potential future work, and conclusions.

II. Communication Growth

The envisioned smart grid network faces unique and specific requirements characterized by the applications present within the grid and their communication needs [4], [13]. Some important types of communication expected on the smart grid's networks include:

- Operational Status Updates – Measurement data published by Synchrophasor Measurement Units (PMUs) throughout the grid
- Control Actions – Remote actions taken to control functioning of smart grid equipment
- Demand Response Messaging – Market information intended to allow consumers to make educated choices about energy use
- Forensics – Engineers investigating unusual behavior

While these applications have a variety of communication needs, one conspicuous attribute is the prevalence of periodic data. Timing and latency of high-rate, periodic data is potentially tightly constrained. Devices also can be expected to communicate for extended periods of time. This in turn shifts emphasis away from setup and handshaking, focusing on the need for efficient, reliable data communication.

III. Benefits of TV-OTS

For years, the power grid has operated with limited wide-area visibility and communication [1]. This is expensive, inflexible, and introduces vulnerabilities. To support the next

This research was funded in part by Department of Energy Award Number DE-OE0000097 (TCIPG). Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

[1] All one-time signature based protocols expire after some period of time. True one-time signature schemes expire after a single message [6]–[8]. Extended schemes can sign multiple messages before expiring [9]–[12]. Once the message limit is reached, the protocol must be restarted with new keys.

generation grid, a communication network needs to be designed for grid-specific applications. Such a network must support the following properties:

- Real-time – Communication delays must be minimal, keeping latency low even over large geographic areas.
- Multicast – Large quantities of data will be intended for multiple recipients. Multicast protocols eliminate the need to overburden the network with duplicate messages.
- Dynamic – Communicating parties should be easy to add and remove from the network.
- Secure – Good encryption and authentication mechanisms must be available to protect sensitive data.

Many authentication protocols [9]–[12], [14]–[16] conflict with the smart grid's needs, restricting communication. TV-OTS supports all current requirements of the smart grid.

1) Multiple Receivers: One of the most important characteristics of TV-OTS is the support for multiple receivers. This feature is extremely important in multicast and broadcast communication. Multicast capabilities enable senders to send the same signed message to all receivers instead of sending a uniquely signed copy to each receiver. Multicast also alleviates the need for senders to be aware of the recipients' identities, simplifying the setup and sending processes.

2) Flexibility: TV-OTS supports an adjustable trade off between security and signature generation and verification overhead. This is a good fit for the power grid. Data generated in large volumes at high rates need inexpensive signatures, but are likely to be less critical than lower rate messages. Lower rate applications are may have more computational resources, allowing more intense signing and verification processes.

3) Latency: TV-OTS relies primarily on one-way functions for security, which are in general inexpensive to compute. Even with large quantities of one-way function evaluations, signatures can be computed more quickly than with other techniques such as RSA [14], [17]. The verification process is low latency as well, with a computational burden roughly analogous to the signing process.

4) Message Independence: Messages signed with TV-OTS are independent, meaning receivers do not rely on the contents of one message to verify others. Independent messages can be verified as soon as they are received, making TV-OTS particularly suitable for real-time systems. Additionally, received packets are immune to the effects of lossy networks.

5) Correspondence with Rate-Based Communication: Signing and verification costs in TV-OTS depend on the anticipated maximum message rate. For data streams operating at fixed rates, the maximum and actual rates are the same. Thus, the associated overhead costs can be reduced to operate at peak efficiency in correspondence with the sending rate.

6) Contextual Awareness: The adjustable security allows external characteristics such as adversaries' assumed computational power and the physical network properties to be taken into account. Security levels are relative to the computational power of a particular adversary and can be adjusted to protect against a specific computational ability. Network characteristics such as expected delay and clock synchronization error are taken into account, preventing adversaries from exploiting the network to gain advantage.

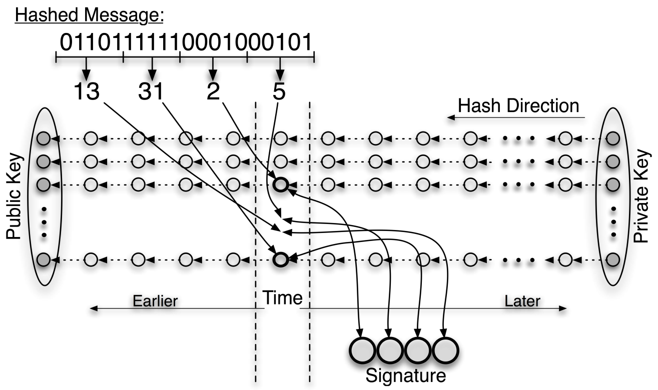

Fig. 1. A small example of the TV-OTS mechanism with 32 hash chains. Hash chains are shown horizontally with individual keys represented by circles. Private and public keys contain the relevant endpoints of the chains. Arrows between chain keys represent derivation by a one-way function. By following the leftward arrows, senders can access any key from any chain. Receivers can only verify a received key by starting at the given key and following the same arrows until a known value is found. Messages are signed and verified by using the message contents and current time to choose hash chains and either retrieve or verify keys. Inclusion of the timestamp in the signature is not shown.

IV. TV-OTS Overview

As described by Wang et al., TV-OTS merges two authentication techniques: one-time signatures and time validation [5]. The resulting protocol combines the strengths of both techniques. Thus, TV-OTS can be viewed as a protocol within a protocol. Individual signatures are created with the Hash of Random Subsets (HORS) [11] protocol, with time validation allowing TV-OTS to remain secure over extended periods of time by periodically refreshing secret keys. The time windows between refreshes are fixed periods referred to as *epochs*. The new secret keys used in each epoch originate from hash chains [18], which allow new secret keys to be validated without needing to transmit new public keys.

Figure 1 shows the mechanism behind signing and verifying in TV-OTS. Each message is hashed and the hash is split into bit strings which are reinterpreted as integer indices. A set of hash chains, indexed by the range of possible indices, is available to supply keys for signatures: generated indices specify the corresponding subset of hash chains which generate keys for the signature. Signers use the hash chains to supply keys to be contained in each signature while receivers use the same set to verify received keys. The current epoch determines which key is selected from each chain. As keys are included in signatures they become public information, increasing the chances of signature forgery by an adversary. To compensate, verifiers use messages' timestamps to ensure received messages are fresh enough that forgery is unlikely.

V. Security Analysis

The adjustable security level is advantageous for adapting to various applications, but vulnerable to misuse. Parameters for TV-OTS are chosen based on an acceptable level of risk. Correct and thorough analysis is necessary to ensure the expected security. Our analysis focuses on the probability of a successful brute force attack, which motivates secure parameter choices[2]. We assume the attacker has eavesdropped a number of exposed keys and is attempting to forge signatures using these eavesdropped keys. Two concerning outcomes are discussed. The first is an attacker forging the signature of a given message m_g. The second is an attacker finding an arbitrary message m_a for which a signature can be forged. The probability of this latter scenario is significantly higher. We analyze both, keeping in mind that an effective attack could be a combination of the two.

Two simplifying assumptions make our analysis worst case:

1) *Exposed keys are all distinct.* In reality, some duplicate keys may be exposed. Publicizing distinct keys increases the number of signatures an attacker can create.
2) *The maximum number of messages are exposed at the beginning of the epoch.* The entire duration of the epoch is then available for attackers to attempt their attacks. This gives attackers more compute time than if messages are sent at intervals throughout the epochs.

The threat of successful attack is independent of epoch. Thus, our analysis spans only a single epoch. To simplify analysis, one epoch is taken as the basic time unit. Variable names used throughout the analysis are given in Table I.

TABLE I
VARIABLE NAMES AND DESCRIPTIONS

N	Number of hash chains
k	Number of keys per message signature
r	Transmitted messages per epoch
h_l	Hash output length
x	Rate at which attackers can check hash collisions

A. Successfully Forging a Given Message

We assume the attacker has a specific message m_g and wishes to forge the signature for m_g. To successfully forge this signature, the particular keys needed to sign m_g must have been exposed. The probability that these keys are exposed depends on the ratio between exposed keys and the total (both exposed and unexposed) keys. The number of exposed keys from which the attacker can choose is rk. These combine into $(rk)^k$ possible signatures. The total number of possible signatures is N^k. Assuming the mapping between messages and signatures distributes messages evenly[3], the probability p_{k_g} of a signature requiring k_g keys being forged is given by:

$$p_{k_g} = (\tfrac{rk}{N})^{k_g} \tag{1}$$

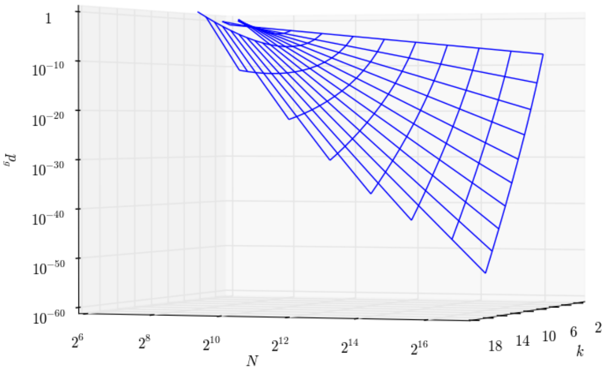

Fig. 2. Behavior of p_g (Equation 3) with N and k with $r = 30$ and $h_l = 160$. Regions where N and k conflict are excluded.

Combinatorics can be applied to find the probability P_{k_g} that exactly k_g different keys are required for a signature. For each choice of k_g distinct keys, we count the number of ways to overlap the remaining $k-k_g$ choices with the already chosen k_g keys. Dividing by the total number[4] of signatures gives:

$$P_{k_g} = \frac{\binom{n}{k_g} \times \binom{k-1}{k_g-1}}{\binom{k+n-1}{n-1}} \tag{2}$$

Thus, the expected p_g of finding a signature for m_g is:

$$p_g = \sum_{k_g=1}^{k} P_{k_g} \times p_{k_g} \tag{3}$$

The values chosen for k and N must be feasible for generating signatures. To avoid impossible signatures, certain constraints must be placed on k and N. Equation 4 prevents uneven mapping of messages to hash chains[5]. Equation 5 ensures the number of bits used to index keys is bounded by the size of the hash output.

$$N = 2^l, \qquad \text{for some } l \tag{4}$$

$$k \times lg_2(N) \leq h_l \tag{5}$$

Optimal choices for N and k are not immediately obvious, especially when constrained by application requirements. Figure 2 gives insight into the behavior of p_g with varying values of N and k. This figure shows that increasing N leads to greater security, but the optimal value of k depends on N.

B. Successfully Finding a Message to Forge

The probability of an attacker finding an arbitrary message to forge takes into account x, the rate at which the attacker can search for m_a. This probability, p_f, is the complement of the probability that in all the attacker's attempts, not one yields a message that can be forged. The probability p_f is given by:

$$p_f = 1 - (1 - p_g)^x \tag{6}$$

[2] Analysis of attacks (brute force, dictionary, DoS, delay and drop packet, and replay attacks) is given by Wang et al. [5], however, the analysis of brute force attacks is critical for choosing secure parameters.

[3] We assume the chosen hash function is strong and produced hashes are indistinguishable from random numbers.

[4] Unlike the number of possible signatures used for Equation 1, the combinatoric analysis of Equation 3 disregards permutations.

[5] A more in depth analysis is required when N is not a power of two.

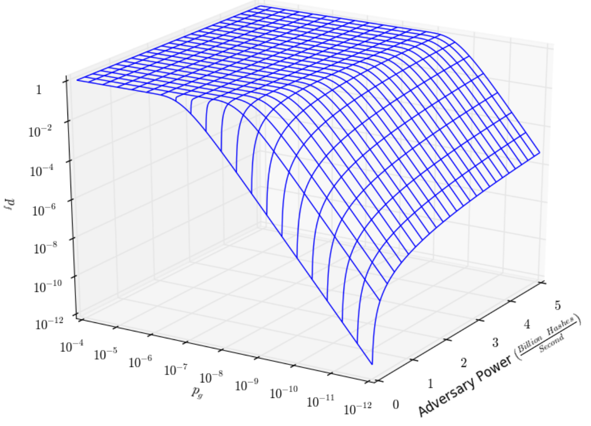

Fig. 3. Behavior of p_f with adversarial power and p_g.

Figure 3 shows the difficulty in achieving a low p_f, even for low values of p_g. In anticipation of powerful attackers, p_g must be set very low to achieve a low p_f. One realistic form of attack utilizes GPU clusters. In an experiment measuring the hashing speed of GPUs, Marks et al. were able to accomplish more than 1.4 billion hashes/second [19]. With constantly evolving technology, we expect realistic attackers to have even higher computational power. Table II gives values of p_g and p_f anticipating an attacker capable of 2 billion hashes/epoch. This table shows the number of hash chains necessary to achieve small p_f values at 30 messages/epoch.

VI. Implementation

This section briefly discusses some choices and implementation details affecting the security and performance of TV-OTS.

A. Hash Chains

Hash chains are, in general, a much studied topic and can be costly if handled naively. With all hash chains, a trade off occurs between storage space and the computational complexity of retrieving keys from already generated chains. The simplest way to optimize key retrieval is to store all keys, but the simplest way to optimize storage space is the opposite: store only the seed and recompute keys as needed. Neither approach is satisfactory, especially if devices are resource constrained. More advanced strategies strive to balance storage and computation costs [15], [20]–[23]. Our implementation uses Fractal Hash Sequence Representation and Traversal (FHT) [24], which supports logarithmic scaling for both storage and computation.

TABLE II
Probabilities calculated for some commonly expected examples scenarios.

Examples with $h_l = 160$, $r = 30$, $x = 2\text{E}9$					
	N, k				
	1024,13	2048,14	4096,13	8192,12	16384,11
p_g	4.5E-6	3.3E-10	7.5E-14	7.3E-17	3E-19
p_f	≈ 1	0.48	1.5 E-4	2.2E-7	≈ 0

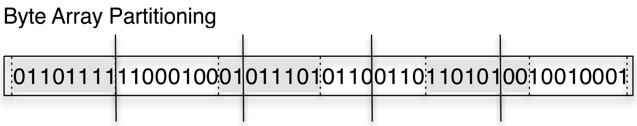

Fig. 4. A minimal example where eight bytes must be split to create four integers with 12 bits each. The leftmost eight bits are discarded.

B. Indexing Function

The indexing function is responsible for splitting message hashes into the indices used to select keys for signatures. If not implemented carefully, this operation can bias the distribution of chosen keys. Each index requires enough bits to index the entire set of hash chains. Each bit should contribute to at most one index to ensure independence between indices. Thus, if $N = 2^l$ hash chains are available, then each index should contain exactly l bits. With k keys included in each signature, the total number of contributing bits will be $k \times l$. Since the number of available bits is limited by the size of the message hash, a trade off is imposed between N and k.

Our function iteratively partitions the message hash into sections of exactly l bits, as shown in the example in Figure 4. This may require splitting individual bytes from the hash: masking and shifting operations are used to ensure the correct bits are used for each integer. If $k \times l$ is smaller than the number of available bits, the remaining bits are ignored and do not contribute to any of the indices.

C. Implementation Details

Our Java implementation provides "bump in the wire" authentication for existing client applications. Clients instantiate authentication objects on either end of a data stream. An initialization parameter specifies whether each instance acts as a signer or verifier. Signing instances take in raw messages and return the message appended with a TV-OTS signature. Verifying instances strip the signatures from signed messages, returning the original message or throwing an error if the message is unverifiable. Once initialized, each authentication instance performs only one of these two functions.

A secondary module encapsulates all hash chain related activity. Authentication instances rely on the hash chain module for supply and verification of hash chain keys, and client applications can query the hash chain module for public and private keys with which to initialize senders and receivers. Similarly to the authentication module, hash chains act as either a supplier or verifier based on an initialization parameter. Supply chains use the storage and retrieval scheme of FHT. Verifying chains only store two values: the original commitment key and the most recently received key. Newly received keys are verified by creating a match with one of these stored keys.

D. TV-OTS In GridStat

TV-OTS was tested within GridStat, a low-latency data delivery system designed and built for the smart grid [25],

TABLE III

LATENCY DATA AND APPROXIMATE SECURITY PROBABILITIES

Chain Length	Number Chains (N)	Keys per Signature	Signed Bits	Epoch (seconds)	Lifetime (h:mm:ss)	Latency (milliseconds)	St. Dev. (σ) (milliseconds)	p_g	p_f
8192	1024	13	130	0.25	0:34:08	**4.39**	1.97	5.3E-14	2.6E-5
				1	2:16:32	**1.87**	0.72	3.6E-6	1
	2048	14	154	0.5	1:08:16	**3.93**	1.51	1.4E-14	1.4E-5
				1	2:16:32	**2.72**	1.03	2.3E-10	0.37
	4096	13	156	0.75	1:42:24	**4.63**	1.91	1.3E-15	1.8E-6
				1	2:16:32	**3.99**	1.52	5.3E-14	1E-4
				1.25	2:50:40	**3.22**	1.17	3.1E-12	0.0077
	8192	12	156	1	2:16:32	**5.5**	2.81	5.2E-17	0
				1.5	3:24:48	**3.87**	1.43	6.7E-15	2E-5
				2	4:33:04	**3.05**	1.1	1.2E-13	8.5E4
	16384	11	154	1	2:16:32	**6.98**	3.44	1.3E-20	0
				3	6:49:36	**2.59**	0.94	6.7E-13	4E-6
				4	9:06:08	**2.1**	0.81	2.1E-13	0.0017
2048 Bit RSA						**42.53**	2.92	—	

[26]. GridStat is a publish subscribe system showcasing low overhead and routing costs. One of the features of GridStat is rate-based communication, where messages are published at fixed rates. TV-OTS was used to authenticate messages flowing between GridStat Publishers and Subscribers. Latency data was gathered by comparing the timestamps collected immediately before signing and after verification and averaged over 10 000 messages. Tests were completed with a single publisher and subscriber running on the same host device to eliminate measurement errors introduced by separate clocks.

The hash algorithm used was SHA-1. The chosen parameters anticipate an adversary capable of 2 billion SHA-1 hashes per second. The sending rate for all tests was 30 messages/second. Latency results are shown in Table III with RSA shown for comparison. Results show a substantial performance increase when comparing to RSA as a benchmark.

Table III provides insight into the latency behavior of TV-OTS. Decreasing epoch duration and increasing N both lead to higher latencies, but a large N combined with a long epoch can reach latencies comparable to smaller N's with short epochs. Instances using 1024 chains with 0.5-second epochs and instances with 16 384 chains and 4-second epochs both have latencies around 2 milliseconds. Tables such as this one can be used to estimate parameter sets meeting certain latency or security requirements.

VII. REAL WORLD SECURITY

The security analysis in Section V gives a theoretical analysis of forgery probabilities, but these measures may not be good indicators of real world security. Some arguments are presented here which would mean better security performance, but may be application specific and require further study.

A. Worst Case Analysis Too Harsh

The assumptions imposed on the theoretical analysis do not represent actual use; primarily, keys are exposed more slowly than assumed. Recall that security decreases with increased key exposure. In real situations, some overlap can be expected between exposed keys, decreasing the number of exposed keys and increasing expected security. Even more importantly, the expectation that all revealed keys will be exposed at the beginning of each epoch is invalid in rate-based systems. New keys are exposed regularly, giving attackers a continually evolving number of keys with an upper bound of rk.

B. Forged Signatures Not Sufficient

The time validation mechanisms make forging a message more difficult than merely gathering keys to forge an arbitrary bit string. The signed message must contain a valid timestamp and sequence number. Thus, even if a message is found for which a signature can be forged, the message itself may not be verifiable. In actuality, the chances of forging a message reduces to the chances of finding a *verifiable* message with a signature that can be forged.

C. Resistance to Bad Data

Smart grid related applications are likely to have a certain amount of robustness against bad data. For example, applications using PMU measurements must be able to account for measurement errors such as those caused by faulty sensors. Applications with well defined communication protocols will require well formed messages. It is likely that an attacker, even forging a small number of arbitrary messages, will not be able to affect the larger context of the grid.

VIII. FUTURE WORK

Beyond continued security analysis, future efforts should consider practical aspects of implementing TV-OTS, such as managing the vast amount of necessary key material.

A difficulty arising from the key structure is that of efficiently bootstrapping receivers. TV-OTS allows receivers to join a message stream at any time. However, the latency of verifying messages is increased for late joining receivers. This stems from the structure of the hash chains and the corresponding public keys. Received keys are authenticated by hashing to create a match with a known key. Receivers save the latest keys they receive, lowering the cost of verifying new keys by shortening the amount of time needed to create a match. Late joining receivers will require time to accumulate a store of known keys, and until then, will need to verify keys by performing enough hash operations to recreate a value in the public key. This problem could be resolved by building intelligence into the key server to update public keys with each epoch. Depending on the location of the key server and the anticipated variability in receivers, such a service may be valuable.

Hash chains cause inherently expensive public key generation. Creating public keys requires enumerating the entire set of hash chains, which is best computed offline due to high computational costs. Fortunately, this operation is easily parallelized and can be computed on devices more powerful than the publishers. If this is not an option, key generation is potentially amortized over epochs of prior protocol instances.

Even with complications accounted for, the large amount of required key material is still a concern. The ratio of generated key material to sent messages suggests that alternative one-time signatures might be a better option than HORS. Alternative signatures, including true one-time signatures, might offer better security with lower key-storage overhead. However, latency testing would need to be factored into such comparisons.

IX. Conclusion

This paper presents an extended analysis of Time-Valid One-Time-Signature, a low-latency multicast data authentication protocol designed for use in the smart grid. TV-OTS fits well in the context of the smart grid due to its highly parameterizable nature. However, the flexibility of TV-OTS adds complications. Parameters must be set carefully to achieve the desired security. These same parameters affect overhead, lowering performance when security is increased. A detailed security analysis shows that high security levels are associated with a large requirement in the amount of key material. This in turn presents practical complications in implementation.

Fortunately, TV-OTS exhibits low latency signing, even under secure parameter sets. TV-OTS was implemented and tested in the GridStat system, a real-time data delivery system for the smart grid. Testing reveals latencies lower than 10 milliseconds for combined signing and verification, which is favorable over the latencies of RSA. Additionally, multiple arguments were presented which suggest real-world factors may improve security of TV-OTS beyond what the worst-case analysis shows. Together, these results suggest that TV-OTS is a good candidate for data authentication in the smart grid.

References

[1] U.S.-Canada Power System Outage Task Force. (2004), “Final Report on the August 14th, 2003 Blackout in the United States and Canada,” online, 2004. [Online]. Available: https://reports.energy.gov

[2] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, R. Schulz, A. Stankovic, C. Taylor, and V. Vittal, “Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance,” *IEEE Trans. Power Systems*, vol. 20, no. 4, pp. 1922–1928, 2005.

[3] R. N. Anderson, “Building the Energy Internet,” *Economist*, 2004.

[4] D. Bakken, A. Bose, C. Hauser, D. Whitehead, and G. Zweigle, “Smart Generation and Transmission With Coherent, Real-Time Data,” *Proc. IEEE*, vol. 99, no. 6, pp. 928 –951, June 2011.

[5] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, “Time Valid One-Time Signature for Time-Critical Multicast Data Authentication,” in *INFOCOM 2009*. IEEE, 2009, pp. 1233–1241.

[6] W. Diffie and M. Hellman, “New Directions in Cryptography,” *IEEE Trans. Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[7] L. Lamport, “Constructing Digital Signatures from a One-Way Function,” CSL-98, SRI International, Tech. Rep., 1979.

[8] R. C. Merkle, “A Certified Digital Signature,” in *Proc. Advances in Cryptology–CRYPTO89*. Springer, 1990, pp. 218–238.

[9] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, “Efficient Authentication and Signing of Multicast Streams over Lossy Channels,” in *Proc. 2000 IEEE Symp. Security and Privacy*. IEEE, 2000, pp. 56–73.

[10] A. Perrig, “The BiBa One-Time Signature and Broadcast Authentication Protocol,” in *Proc. 8th ACM Conf. Computer and Communications Security*. ACM, 2001, pp. 28–37.

[11] L. Reyzin and N. Reyzin, “Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying,” in *Information Security and Privacy*. Springer, 2002, pp. 1–47.

[12] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, “The TESLA Broadcast Authentication Protocol,” *RSA CryptoBytes*, vol. 5, no. 2, 2002.

[13] P. Kansal and A. Bose, “Bandwidth and Latency Requirements for Smart Transmission Grid Applications,” *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1344–1352, 2012.

[14] R. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[15] G. Itkis and L. Reyzin, “Forward-Secure Signatures with Optimal Signing and Verifying,” in *Advances in Cryptology–Crypto 2001*. Springer, 2001, pp. 332–354.

[16] A. Perrig, R. Canetti, D. Song, and J. Tygar, “Efficient and Secure Source Authentication for Multicast,” in *Network and Distributed System Security Symposium, NDSS*, vol. 1, 2001, pp. 35–46.

[17] C. Hauser, T. Manivannan, and D. Bakken, “Evaluating Multicast Message Authentication Protocols for Use in Wide Area Power Grid Data Delivery Services,” in *2012 45th Hawaii Int. Conf. System Science (HICSS)*. IEEE, 2012, pp. 2151–2158.

[18] L. Lamport, “Password Authentication with Insecure Communication,” *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[19] M. Marks, J. Jantura, E. Niewiadomska-Szynkiewicz, P. Strzelczyk, and K. Góźdź, “Heterogeneous GPU & CPU Cluster for High Performance Computing in Cryptography,” *Computer Science*, vol. 13, no. 2, pp. 63–79, 2012.

[20] D. Coppersmith and M. Jakobsson, “Almost Optimal Hash Sequence Traversal,” in *Financial Cryptography*. Springer, 2003, pp. 102–119.

[21] Y. Sella, “On The Computation-Storage Trade-Offs of Hash Chain Traversal,” in *Computer Aided Verification*. Springer, 2003, pp. 270–285.

[22] S. Kim, “Improved Scalable Hash Chain Traversal,” in *Applied Cryptography and Network Security*. Springer, 2003, pp. 86–95.

[23] D. Yum, J. Seo, S. Eom, and P. Lee, “Single-Layer Fractal Hash Chain Traversal with Almost Optimal Complexity,” *Topics in Cryptology–CT-RSA 2009*, pp. 325–339, 2009.

[24] M. Jakobsson, “Fractal Hash Sequence Representation and Traversal,” in *Proc. 2002 IEEE Int. Symp. Information Theory*. IEEE, 2002, p. 437.

[25] D. E. Bakken, C. H. Hauser, H. Gjermundrød, and A. Bose, “Towards More Flexible and Robust Data Delivery for Monitoring and Control of the Electric Power Grid,” Tech. Rep., 2007.

[26] H. Gjermundrod, D. E. Bakken, C. H. Hauser, and A. Bose, “GridStat: A Flexible QoS-Managed Data Dissemination Framework for the Power Grid,” *IEEE Trans. Power Delivery*, vol. 24, no. 1, pp. 136–143, 2009.