Multicast Delayed Authentication for Streaming Synchrophasor Data in the Smart Grid

Sérgio Câmara, Dhananjay Anand, Victoria Pillitteri, Luiz Carmo

▶ **To cite this version:**

Sérgio Câmara, Dhananjay Anand, Victoria Pillitteri, Luiz Carmo. Multicast Delayed Authentication for Streaming Synchrophasor Data in the Smart Grid. 31st IFIP International Information Security and Privacy Conference (SEC), May 2016, Ghent, Belgium. pp.32-46, 10.1007/978-3-319-33630-5_3 . hal-01369539

HAL Id: hal-01369539

https://inria.hal.science/hal-01369539v1

Submitted on 21 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Multicast Delayed Authentication For Streaming Synchrophasor Data in the Smart Grid

Sérgio Câmara[1]*, Dhananjay Anand[2],
Victoria Pillitteri[2], and Luiz Carmo[1]

[1] National Institute of Metrology, Quality and Technology
Duque de Caxias, 25250-020, Rio de Janeiro, Brazil
{smcamara,lfrust}@inmetro.gov.br
[2] National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
{dhananjay.anand,victoria.pillitteri}@nist.gov

Abstract. Multicast authentication of synchrophasor data is challenging due to the design requirements of Smart Grid monitoring systems such as low security overhead, tolerance of lossy networks, time-criticality and high data rates. In this work, we propose *inf*-TESLA, Infinite Timed Efficient Stream Loss-tolerant Authentication, a multicast delayed authentication protocol for communication links used to stream synchrophasor data for wide area control of electric power networks. Our approach is based on the authentication protocol TESLA but is augmented to accommodate high frequency transmissions of unbounded length. *inf*-TESLA protocol utilizes the Dual Offset Key Chains mechanism to reduce authentication delay and computational cost associated with key chain commitment. We provide a description of the mechanism using two different modes for disclosing keys and demonstrate its security against a man-in-the-middle attack attempt. We compare our approach against the TESLA protocol in a 2-day simulation scenario, showing a reduction of 15.82% and 47.29% in computational cost, sender and receiver respectively, and a cumulative reduction in the communication overhead.

Keywords: Multicast authentication, Smart Grid, synchrophasors, Wide Area Monitoring Protection and Control

1 Introduction

Smart Grids are large critical cyber-physical infrastructures and are being transformed today with the design and development of advanced real-time control applications [11]. The installation of Phasor Measurement Units (PMUs) as part of world-wide grid modernization is an example of major infrastructure investments that require secure standards and protocols for interoperability [1].

PMUs take time-synchronized measurements of critical grid condition data such as voltage, current, and frequency at specific locations that are used to

* This work is supported in part by grants from H2020 EU-BR SecureCloud (Grant No. 2568).

provide wide area visibility across the grid.The synchrophasor data aggregated from multiple PMUs are used to support real-time analysis, planning, corrective actions, and automated control for grid security and resiliency. Currently, high-speed networks of PMUs are being used for Wide Area Monitoring Protection and Control (WAMPAC) applications to provide situational awareness in the Eastern and Western Interconnection of North America, in China, Canada, Brazil and across Europe [11]. Before the installation of PMUs, the lack of wide-area visibility is one of the factors that prevented early fault identification of the 2003 Northeast America and 2003 Italy blackouts [21] [9]. Malicious PMU data or deliberate attacks could result in inaccurate decisions detrimental to grid safety, reliability, and security, that said, PMUs need information authentication and integrity, while confidentiality may be considered optional.

Authentication schemes in the Smart Grid must be able to efficiently support multicast. Current standard solution, suggested by IEC 62351 [5], comprises HMAC authentication algorithm for signing the synchrophasors. However, sharing only one symmetric key across a multicast group cannot guarantee adequate security, and this approach suffers from the scalability problem. The use of asymmetric cryptography and digital signatures for multicast authentication raises concerns about the impact on cost and microprocessor performance. One-Time Signature schemes can enable multicast authentication, however they suffer from communication and storage overhead, and complicated key management [24].

Although some previous literature works assume, in general, that delayed authentication is not suitable for real-time applications [7] [8], such method is still eligible for some monitoring and control applications that permit relatively larger delay margins (e.g. wide-area oscillation damping control application) [25]. For more considerations on this topic, see Section 2. Moreover, delayed authentication presents advantages over cited issues by supporting multicast data streaming, symmetric and lightweight cryptography, corrupt data and attack detection. Also it allows scalable solutions and key management, tolerates packet loss, and provides low communication overhead and high computational efficiency.

The primary objective of this work is to propose a multicast delayed authentication protocol called *inf*-TESLA in order to provide measurement authentication in a WAMPAC application within the Smart Grid. Also, we design the Dual Offset Key Chains mechanism which is used by our protocol to generate the authenticating keys and to provide long-term communication without the need of key resynchronization between the sender and receivers. A description of two different modes for disclosing keys and a demonstration of a man-in-the-middle attack attempt against our mechanism are also provided.

Section 2 presents an overview of the network architecture used for wide area aggregation of PMU data as well as some delay constraints and authentication infrastructure. In Section 3 we discuss prior work in the area of packet based authentication protocols for streaming communication, and then in Section 4 we present the *inf*-TESLA protocol and describe the Dual Offset Key Chains mechanism along with its security properties and conditions. In Section 5 we evaluate

our approach against the original TESLA protocol. Finally, we summarize our results and propose future works in Section 6.

2 Scenario Characteristics

The network architecture considered for this work is as follows. Each communication link in the infrastructure comprises one PMU sender node S capable of multicasting packets to m receivers R_k applications, where $1 \leq k \leq m$. PMU S sends time-stamped synchrophasor data packets at a rate of 10 to 120 packets per second and that can be dropped in the way to the receivers. The network has several n intermediate nodes between S and $R_k, n > 0$, called Phasor Data Concentrators (PDCs). PDCs can chronologically sort received synchrophasors as well as aggregate, repackage and route data packets to the set of higher level PDCs (Super PDCs). When packets are missing or lost, PDCs may (with due indication) interpolate measurements in order to retain the communication link.

There are different wide-area monitoring and control applications that consume synchrophasor data and have different time delays and quality requirements. For instance, Situational Awareness Dashboard, Small-Signal Stability Monitoring, and Voltage Stability Monitoring/Assessment accept up to 500 milliseconds in communication latency, other applications such as Long-term stability control, State Estimation, and Disturbance Analysis Compliance can handle up to 1000 ms. For the entire list, see [20].

Zhu *et al.* [25] simulates the latency for monitoring applications over the Smart Grid network architecture and obtained results within a range of 150–220 ms. For centralized control applications, the latency was well below 500 ms. From the delayed authentication perspective, the minimum delay of the authentication confirmation by R_k is approximately twice the latency of the network. Still, delayed authentication protocols are able to attend the requirements for the above cited applications.

When utilizing multicast communication, IEC 61850-90-5, the standard for communication networks and systems for power utility automation, requires a Key Distribution Center (KDC), which provides the symmetric key coordination between S and R_k. We assume that each S is its own KDC, which is also endorsed by the standard. Furthermore, as our scheme demands that S prove its identity to R_k once during communication initialization, each receiver is required to validate a digital signature from S and maintaining a copy of its public key certificate. For this purpose, we assume that a Public-Key Infrastructure (PKI) is also available.

2.1 Security Considerations

We assume that attacks are accordingly aligned, via a man-in-the-middle, to either manipulate data values or masquerade as a legitimate PMU. Using the attack model from [23], the adversary is not limited by network bandwidth and has full control to drop, resend, capture and manipulate packets. Although his

computational resources can be large, it is not unbounded and he cannot invert a pseudorandom function with non-negligible probability. Each receiver R_k is able to authenticate both the content and source of synchrophasor payloads after a delay of d_{NMax} using our delayed authentication scheme presented in Section 4. However, if a packet fails authentication at time t, then an attack that has been active and undetected since $t - d_{NMax}$ represents the maximum threat exposure.

The security primitives used throughout this paper are as follows:

- *One-way hash function* H operates on an arbitrary length input message M, returning $h = H(M)$. H can be implemented with SHA-2 family algorithms.
- *Message Authentication Code* $MAC(K, M)$ provides a tag that can verify authenticity and integrity of message M given a shared key K. $HMAC(K, M)$ is a specific construction which includes an underlying cryptographic hash function to create the authenticating tag.
- *Hash chain* $H^n(M)$ denotes n successive applications of cryptographic hash function H to message M.

3 Related Work

Multicast authentication is an active research field in recent years and has been applied to a wide range of applications. In Smart Grids, it is being used for monitoring, protection and information dissemination [24]. In this section, we review all the TESLA-based multicast authentication schemes and other multicast authentication schemes used for electrical power systems.

To address the challenge of continuous stream authentication for multiple receivers on a lossy network, Timed Efficient Stream Loss-tolerant Authentication (TESLA) was introduced by Perrig et al [14]. Based on the Guy Fawkes protocol [2] and requiring loose time synchronization between the senders and receivers, TESLA is a broadcast authentication protocol considering delayed disclosure of keys used for authentication of previous sent messages and packet buffering by the receiver. This protocol supports fixed/dynamic packet rate and delivers packet loss robustness and scalability. Benefits of TESLA include a low computation overhead, low per-packet communication overhead, arbitrary packet loss is tolerated, unidirectional data flow, high degree of authenticity and freshness of data. Further work proposed several modifications and improvements to TESLA, allowing receivers to authenticate packets upon arrival, improved scheme scalability, reduction in overhead, and increased robustness to denial-of-service attacks [13].

Studer et al. describe TESLA++ [19], a modified version of TESLA resilient to memory-based DoS attacks. They combine TESLA++ and ECDSA signatures to build an authentication framework for vehicular ad hoc networks.

μTESLA [17] adapts TESLA to make it practical for broadcast authentication in severely resource-constrained environments; like sensor networks. Some of these adaptations include the use of only symmetric cryptography mechanisms, less frequent disclosure of keys and restriction on the number of authenticated senders. Liu and Ning [10] reduce the overhead needed for broadcasting key

chain commitments and deal with DoS attacks. Their Multilevel μTESLA protocol considers different levels of key chains to cover the entire lifespan of a sensor.

Other methods include the One-Time Signatures family which gained popularity recently and is applicable to multicast authentication and also for WAMPAC applications. The author in [12] describes a one-time signature based broadcast authentication protocol based on BiBa. BiBa uses one-way functions without trapdoors and exploits the birthday paradox to achieve security and verification efficiency. Its drawbacks include a large public key and high overhead for signature generation.

HORS [18] is described by Reyzin et al. as an OTS scheme with fast signing and signature verification using a cryptographic hash function to obtain random subsets for the signed message and for verifying it, but it still suffers from frequent public key distribution. TSV [8] multicast authentication protocol generates smaller signatures than HORS and has lower storage requirement at the cost of increased computations in signature generation and verification. TSV+ [7], a patched version of TSV, uses uniform chain traversal and supports multiple signatures within an epoch. SCU [22] is a multicast authentication scheme designed for wireless sensor networks and SCU+ [7] adapts it for power systems using uniform chain traversal as well. TV-HORS [23] uses hash chains to link multiple key pairs together to simultaneously authenticate multiple packets and improves the efficiency of OTS by signing the first l bits of the hash of the message. As a downside, TV-HORS has a large public key of up to 10 Kbytes.

4 Proposed Solution

In this section, we propose *inf*-TESLA, a TESLA based scheme. At first, we review TESLA to give some background and then present our scheme.

4.1 TESLA

Timed Efficient Stream Loss-tolerant Authentication (TESLA) [14] [13] [15] [16] is a broadcast authentication protocol with low communication and computation overhead, tolerates packet loss and needs loose time synchronization between the sender and the receivers.

TESLA relies on the delayed disclosure of symmetric keys, therefore the receiver must buffer the received messages before being able to authenticate them. The keys are generated as an one-way chain and are used and disclosed in the reverse order of their generation. At setup time, the sender must first set n as the index of the first element K_n. For generating the key chain, the sender picks a random number for K_n and using a pseudo-random function f, he constructs the one-way function $F : F(k) = f_k(0)$. So, the sender generates recursively all the subsequent keys on the chain using $K_i = F(K_{i+1})$. By that, the last element of the chain is $K_0 = F^n(K_n)$, and all other elements could be calculated using $K_i = F^{n-i}(K_n)$.

Each K_i looks pseudo-random and an adversary is unable to invert F and compute any K_j for $j > i$. In the case of a lost packet containing K_i, a receiver can calculate K_i given any subsequent packet containing K_j, where $j < i$, since $K_j = F^{i-j}(K_i)$. As a result, TESLA tolerates sporadic packet losses.

The stream authentication scheme of TESLA is secure as long as the security condition holds: A data packet P_i arrived *safely*, if the receiver can unambiguously decide, based on its synchronized time and maximum time discrepancy, that the sender did not yet send out the corresponding key disclosure packet P_j.

TESLA also supports both communication with fixed or dynamic packet rate. For fixed rate, the sender discloses the key K_i of the data packet P_i in a later packet P_{i+d}, where d is a delay parameter set and announced by the sender during setup phase. The sender determines the delay d according to the packet rate r, the maximum tolerable synchronization uncertainty δ_{tMax} and the maximum tolerable network delay d_{NMax}, setting $d = \lceil(\delta_{tMax} + d_{NMax})r\rceil$. In this mode, the scheme can achieve faster transfer rates. For dynamic rate, the sender pick one key per time interval T_{int}. Each key is assigned to a uniform interval of duration T_{int}, T_0, T_1 ,..., T_n, that is, key K_i will be active during the time period T_i. The sender uses the same key K_i to compute the MAC for all packets which are sent during T_i, on the other hand, all packets during T_i disclose the key $K_{i-d'}$. In this case, $d' = \lceil(\delta_{tMax} + d_{NMax})/T_{int}\rceil$. We use the designation d and d' for fixed and dynamic rates respectively.

For each new receiver that joins the communication network, the sender initially creates an authenticated synchronization packet. This packet contains parameters such as interval information, the disclosure lag and also a disclosed key value - which is a commitment to the key chain. The sender digitally signs this packet to each new receiver before starting the streaming communication.

4.2 *inf*-TESLA

inf-TESLA, short for infinite TESLA, is a multicast authentication protocol based on TESLA suitable for use in long term communication at high packet rates. As in TESLA, *inf*-TESLA relies on the strength of symmetric cryptography and hash functions and on the delayed disclosure of keys as a means to authenticate messages from the sender. Also, it requires only loose time synchronization between the sender and the receiver and can operate under both dynamic and fixed packet rates.

By using fixed packet rate mode, there is no need for setting specific time intervals for MACing and disclosing keys. Each autheticating key is used once for the actual message and disclosed d packets later. Although this operational mode can achieve maximum speed on authenticating previous packets, it has a drawback of quickly consuming the authenticating key chain, depending on the frequency of the packets.

Since we use one-way hash functions to build independent key chains, every time one of the key chains comes to an end (meaning that it was fully used in the authentication process) the sender must automatically build, store and utilize a new key chain in its place. In the original TESLA protocol, a sender would have

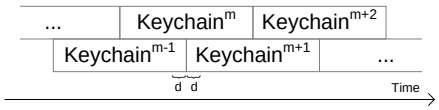

Fig. 1. An illustration of dual offset key chains as used for *inf*-TESLA.

to reassign a new synchronization packet as the current key chain comes to an end, inflicting non-negligible network and computational overhead by digitally signing a synchronization packet at the end of each key chain.

inf-TESLA addresses this issue by using the Dual Offset Key Chains mechanism. This mechanism uses a pair of keys for each message and guarantees continuity of the multicasting authentication process without the need for signing and sending a new synchronization packet. The mechanism creates two offset key chains so that a pair of active key chains are always available and, as the main principle, a key chain m always straddles the substitution of key chain $m-1$ with $m+1$. Figure 1 illustrates the Dual Offset Key Chains mechanism by which key chain m supports the substitution of key chain $m-1$ for key chain $m+1$ without the need for resynchronization. A detailed description of the Dual Offset Key Chains mechanism is presented on Section 4.2.

The overall initialization setup is similar to TESLA. Before the data streaming begins, the sender first determines some fundamental information about the network status, d_{NMax}), and time synchronization, δ_{tMax}, and builds its first two key chains. We assume that both sender and receiver are time synchronized by a reliable time protocol (e.g. PTP). After that, the sender S chooses the delay parameter d (Section 4.1) that will base the decision of the receiver R_k to either accept a packet from S. This condition is **Security Condition-1** for *inf*-TESLA.

For bootstrapping each new receiver, S constructs and sends the synchronization (commitment) packet to the new incomer. For a dynamic packet rate, this packet contains the following data [13]: the beginning time of a specific interval T_j along with its id I_j, the interval duration T_{int}, the key disclosure delay d', a commitment to the key chain K_i^m and key chain K_i^{m+1} ($i < j - d'$ where j is the current interval index).

For a fixed packet rate r, let j_1 and j_2 be the current key from key chains m and $m+1$ respectively. The synchronization packet contains: delay d and the commitment for the key chains $K_{i_1}^m$ and $K_{i_2}^{m+1}$ ($i_1 < j_1 - d$ and $i_2 < j_2 - d$). We will focus on fixed packet rate in this paper for the sake of brevity and convenience of notation. While a fixed packet rate is potentially more likely for the streaming applications we address, our approach is compatible with both dynamic and fixed rates.

Dual Offset Key Chains mechanism. The Dual Offset Key Chains mechanism enables continuity in streaming authentication without the periodic resynchronization between S and $R_k \in R$ required by TESLA. Two key chains, offset

in alignment, are used simultaneously by the mechanism to authenticate messages. For every packet, there are always two active key chains and, from each chain, one non-used key available for MACing.

For constructing the two key chains, first the sender chooses n, the total number of elements on a single key chain. Let l^m be the current number of remaining elements on the key chain m. Here we assume that all created keys are deleted just after being used for authenticating messages. Let M be the maximum available memory for storing the key chains, assuming that M is big enough for storing two key chains, m and $m+1$, at any time. The value of n must be chosen accordingly to the following constraints: (i) $n \geq l^{m-1} + 2(d+1)$ and (ii) $n \leq \frac{M}{2} + d$.

The the first constraint sets a minimum value for n, that is the minimum initial size of a key chain. During the initialization setup of the first receiver synchronization, we consider $l^{m-1} = 0$ for constructing the first key chain. The second constraint restricts the maximum number of elements in a key chain. If a key chain m does not meet this limit, key chain $m+1$ will not be long enough to meet the security condition for the key chain exchange procedure (see Section 4.2). In practice, it may not be feasible to calculate a whole key chain in the time taken to send two data packets and so S may compute and store key chain $m+1$ well before the end of key chain $m-1$.

A packet P_j sent by S is formed by the following data $P_j = \{M_j, i_1, i_2, K^m_{i_1-d}, K^{m+1}_{i_2-d}, MAC(K^m_{i_1}||K^{m+1}_{i_2}, M_j)\}$. Every packet carries the actual message M_j, the current sequence number of each key chain i_1 and i_2, the disclosed authenticating keys $K^m_{i_1-d}$ and $K^{m+1}_{i_2-d}$ (discussed later in Section 4.2) and the MAC of the message resultant from an operation that uses the concatenation of current keys from both key chains. In particular, at the beginning of a key chain $m+1$, the notation $K^{m+1}_{i_2-d}$ may refer to the last keys in the key chain $m-1$.

Disclosure of keys. *inf*-TESLA has two modes of operation for disclosing keys: **2-keys** and **Alternating**. In the 2-keys mode (or standard mode, as previously described), each packet P_j discloses two authentication keys, one from each key chain, for the same message M_i, that is packet P_j has the following information, $P_j \rightarrow K^m_{i_1-d}, K^{m+1}_{i_2-d}$.

The Alternating mode discloses one key from each key chain alternatively in each data packet. Formally, two consecutive packets would have the following information about keys, $P_j \rightarrow K^m_{i_1-d}$ and $P_{j+1} \rightarrow K^{m+1}_{i_2+1-d}$, where indexes $i_1 - d$ and $i_2 - d$ correspond to the keys of both key chains to be disclosed in the same data packet in 2-keys mode of operation. Figure 2 shows the key chains in time and the two modes for disclosing keys. In Section 5, we present a more detailed comparison of these two modes in relation to communication overhead, computational cost and authentication delay.

The disclosure delay d for the keys is directly affected by the maximum tolerable network delay d_{NMax}, so each receiver R_k will present a different delay value. Sender S must set d as the largest expected delay in order to meet security condition-1.

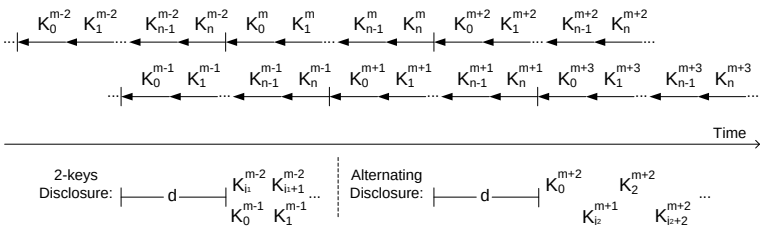

Fig. 2. Two modes for disclosing keys: 2-keys and alternating.

Dual Offset Key Chains mechanism security. Key chain security is based on the widely used cryptographic primitive: the one-way chain. One-way chains were first used by Lamport for one-time password [6] and has served many other applications in the literature.

The **Security Condition-2** for *inf*-TESLA concerns the key chain exchange procedure. This condition states that both key chains cannot be substituted within a time interval d/r (or within d packets). If this happens, the receiver must drop the following packets and request for resynchronization with the sender. This protocol restriction assures the authentication inviolability of *inf*-TESLA and must be observed at all times by the receiver. The receiver is solely responsible for monitoring the key chain exchange procedure and accepting, or rejecting, the new key chain.

In Figure 3, we show an example of a man-in-the-middle attack attempt on the Dual Offset Key Chains mechanism and the importance of the security condition-2. For this example, we consider $d = 9$ as minimum number packets the sender has to wait to disclose a key, the last element $n = 50$ for all key chains, and the asterisk symbol indicates an item maliciously inserted by the attacker. The packets are presented without indices "i" for cleaner presentation.

We first illustrate how this attack can work on a single key chain mechanism without commitment packets as follows: When the attacker senses a change in the key chain by testing every disclosured key *(a)*, he inserts M_0^* as the first manipulated message and MACs it using the first element K_0^* of a forged key chain of his own. The attacker continues faking the messages and its MACs till the last authentic key used for MACing is disclosured. After that point, the attacker is able to take complete control of the communication without being detected (b). For the second part of Figure 3, the same attack is attempted against our mechanism. Also the attacker is able to sense when a disclosured key chain comes to an end and can also substitute the messages and the MACs in the packets. However, when he tries to take complete control of the key chain by forcing the forged key K_0^{**} over the key chain $m = 2$, this indicates for the receiver a violation of the security condition-2 for the key chain exchange procedure.

Another concern is how many consecutive packets could be lost by the receiver without actually being an attack. Following the security condition for key chain substitution, there must not be two different key chain substitutions

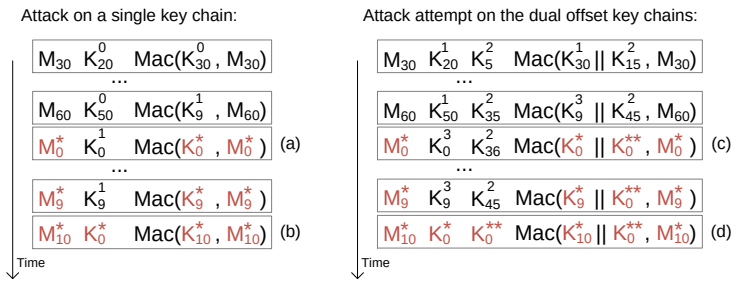

Fig. 3. Example of a man-in-the-middle attack on a single hash chain without commitment and on the Dual Offset Hash Chains mechanism.

within a period of d/r, so the receiver must be aware that the limit for consecutive packets lost is at maximum d. If, for some reason, more than d packets are lost/dropped, the receiver must assure that the following disclosed keys are authentic elements of at least one of the existing key chains, otherwise the receiver will not be able to authenticate any of the next received packets. From this point, the receiver must refuse this stream and request for a new synchronization with the sender.

Another security issue can occur when the last key K_n in the key chain's sequence is lost, that can cause a total lack of authentication of a previous packet P_n. When some K_i is lost, it can be computed from any subsequent key in the key chain through function F (Section 4.1), however when $i = n$ there is no subsequent key. This issue can be extended for the last d elements of the key chain, meaning that in this scenario some packets may not be authenticated and then must be dropped by the receiver. For the Alternating mode for disclosing keys, the receiver would drop d+1 packets in the worst case. This issue concerning the last keys of the key chain is a vulnerability of the original TESLA as well.

Elaborate attacks, like selective drop of packets, can cause even more authentication delay without being noticed. For instance, in the case of the Alternating keys disclosure mode, one attacker can induce an alternating drop of packets preventing the sender to authenticate some sequential packets. To mitigate these attacks, the receiver must set an upper limit for the maximum number of non authenticated packets to ignore before resynchronizing with the sender.

5 Evaluation against TESLA

For the following comparison evaluation, we check for communication overhead, authentication delay and computational cost on a long term communication for each of the following schemes: original TESLA, *inf*-TESLA 2-keys (two disclosured keys per packet) and *inf*-TESLA alt (alternating key chain disclosure). Due to PMUs' operational settings, we are only considering a fixed packet rate mode. Also, we assume the following constraints for the simulation:

Table 1. Communication overhead.

	Formula
TESLA (fixed)	$C * (sKey + sSig) + P * (sKey + sMac)$
Inf-TESLA 2-keys	$2 * sKey + sSig + P * (2 * sKey + sMac)$
Inf-TESLA alt	$2 * sKey + sSig + P * (sKey + sMac)$
	2-day simulation (MBytes)
TESLA (fixed)	331,825
Inf-TESLA 2-keys	497,664
Inf-TESLA alt	331,776

- Phasor data frame size of 60 bytes, according to the C37.118 standard [25], over UDP transport layer protocol.
- HMAC function and f function implementation as HMAC-SHA-256-128. Both HMAC tag size and key size of 128 bits (truncated).
- Digital signature implemented as ECDSA over GF(p) of 256 bits. Although TESLA considers RSA signatures, for comparison purposes we use ECDSA. The keys and signatures sizes are based on the NIST SP 800-131A [3] for recommendations on use of cryptographic algorithms and key lengths.
- Maximum number of keys n that can be stored at a time in the cache memory of a device is 10,000 keys.
- Sender's packet rate (frequency) of 60 packets/sec.
- Simulation time of 2 days. Past references [7] established a baseline of 1024 key chains for evaluating OTS multicast schemes. However, as *inf*-TESLA must build approximately 4 times the number of key chains as TESLA for the same number of packets, comparisons are done for fixed simulation duration rather than number of key chains. Still, for the given constraints, TESLA needs an approximate number of 1024 key chains to operate.

Table 1 shows the formulas to calculate all security related communication overhead of each of the 3 schemes. Let C be the number of commitments (signed packets), P the total number of transmitted packets and $sKey$, $sMac$ and $sSig$ be the size of a cryptographic key, the size of the MAC tag and the size of a signature tag respectively. *inf*-TESLA 2-keys presents the higher communication overhead due to two disclosed keys per packet, while TESLA and *inf*-TESLA alt present a slightly, but negligible, difference on the overhead during two days of operation.

For calculating the computational cost overhead of each scheme, we use the formulas shown in Table 2. The processing cost in cycles per each operation of hashing, macing, signing and verifying is represented by $cHash$, $cMac$, $cSig$ and $cVer$ respectively. From the graph in Figure 4, we can observe the higher computational cost of the sender and receiver operating TESLA over *inf*-TESLA, due to constant signing and verification operations.

For two days of simulation in this configuration, a sender running TESLA protocol on fixed packet rate mode has to sign up to 1036 commitment packets and spends on average 0.373117 gigacycles/hour, while running *inf*-TESLA he would spend 0.314087 gigacycles/hour of operation, which means a reduction of 15.82% in computational cost for the sender. On the receiver side, a TESLA

Table 2. Computational cost calculation.

	Sender
TESLA (fixed)	$C * cSig + P * (cMac + cHash)$
Inf-TESLA (both)	$cSig + 2 * P * (cMac + cHash)$
	Receiver
TESLA (fixed)	$C * cVer + P * (cMac + cHash)$
Inf-TESLA (both)	$cVer + 2 * P * (cMac + cHash)$

receiver spends in average 0.596289 gigacycles/hour, while *inf*-TESLA needs 0.314303 gigacycles/hour, meaning a reduction of 47.29% in computational cost for the receiver. All values of cycles/operation of the security primitives are referenced from the Crypto++ Library 5.6.0 Benchmarks [4].

Although the alternating keys disclosure mode showed good results on the two previous evaluations, this mode increases the authentication delay of a packet P_i by one packet. That is because the second key needed for authenticating P_i, i.e. $K_{i_2}^{m+1}$, will only be disclosed on P_{j+1} where $j > i + d$. Also, if P_{j+1} happens to be lost, the authentication of P_i will be only achieved when receiver has the disclosed key included in P_{j+3}. On both other schemes, the authentication of a packet P_i is normally achieved after receiving P_j, $j > i + d$, and if P_j is lost, the missing keys can be recovered from the contents in P_{j+1}. Also regarding authentication delay evaluation, necessary time overhead for generation and verification of digital signatures during key chains exchange may affect TESLA's continuous flow on higher frequencies of streamed data.

Although TESLA protocol is an efficient protocol and has low security overhead, it was not originally designed for long-term communication at high packet rates. We observe that *inf*-TESLA, in alternating disclosure mode, can deliver a slightly lower communication overhead and, for both modes, result in a sig-

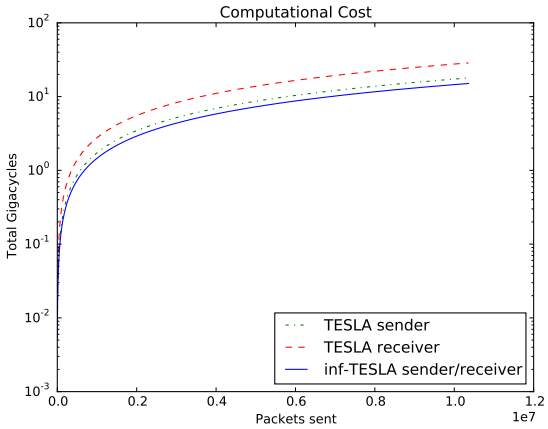

Fig. 4. Computational cost for TESLA and *inf*-TESLA over 2 days of streaming data.

nificant reduction in computational overhead over the original protocol for the given conditions. In general, *inf*-TESLA scheme also provides great suitability for key storage and computational constrained devices, such as in Wireless Sensor Networks (WSNs).

6 Conclusion

In this work, we present *inf*-TESLA, a multicast delayed authentication protocol for streaming synchrophasor data in the Smart Grid, suitable for long-term communication and high data rates scenarios. To authenticate messages from the sender, *inf*-TESLA uses two keys to generate the MAC of the message and discloses both keys after a time frame d/r, on a fixed packet rate of operation.

We also design the Dual Offset Key Chains mechanism to produce the authenticating keys and provide a long-term communication without the need of frequently signing resynchronization packets containing commitments to the new key chains, which ensures continuity of the streaming authentication. We prove our mechanism against a man-in-the-middle attack example and describe the security conditions that must be observed at all times by the receiver. *inf*-TESLA enables two different modes for disclosing keys, 2-keys (or standard) and Alternating keys. We present a comparison between this two modes against TESLA within a WAMPAC application, and our protocol shows even more efficiency when compared to the original. Although the Alternating key disclosure mode increases the authentication delay by one packet, it provides less impact on communication overhead and a reduction of 15.82% and 47.29%, sender and receiver respectively, in computational cost during operational time. Generally, *inf*-TESLA shows promise and suitability for key storage and computational constrained devices.

In future work, we intend to do a further analysis on the trade-off between key storage size in devices and protocol performance, and on the possible (minimum/maximum/average) values for the authentication delay by simulating our protocol in a WAMPAC network.

References

1. Greer et al., C.: NIST Framework and Roadmap for Smart Grid Interoperability Standards. Tech. rep., NIST (2014)
2. Anderson, R., Bergadano, F., Crispo, B., Lee, J.H., Manifavas, C., Needham, R.: A new family of authentication protocols. ACM SIGOPS Operating Systems Review 32, 9–20 (1998)
3. Barker, E., Roginsky, A.: Recommendation for transitioning the use of cryptographic algorithms and key lengths. SP 800-131A Transitions (2011)
4. Dai, W.: Crypto++ 5.6. 0 benchmarks. Website at http://www.cryptopp.com/benchmarks.html (2009)
5. International Electrotechnical Commission: IEC TS 62351-1 Power systems management and associated information exchange - Data and communications - Part 1: Communication network and system security-Introduction to security issues (2007)

6. Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24(11), 770–772 (1981)
7. Law, Y.W., Gong, Z., Luo, T., Marusic, S., Palaniswami, M.: Comparative study of multicast authentication schemes with application to wide-area measurement system. Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security p. 287 (2013)
8. Li, Q., Cao, G.: Multicast authentication in the smart grid with one-time signature. IEEE Transactions on Smart Grid 2, 686–696 (2011)
9. Liscouski, B., Elliot, W.: Final report on the August 14, 2003 blackout in the United States and Canada: Causes and recommendations. A report to US Department of Energy 40(4) (2004)
10. Liu, D., Ning, P.: Multilevel μTESLA: Broadcast Authentication for Distributed Sensor Networks. ACM Trans. Embed. Comput. Syst. 3, 800–836 (2004)
11. Patel, M., Aivaliotis, S., Ellen, E.: Real-time application of synchrophasors for improving reliability. NERC Report, Oct (2010)
12. Perrig, A.: The BiBa one-time signature and broadcast authentication protocol. Proceedings of the 8th ACM conference on Computer and Communications Security p. 28 (2001)
13. Perrig, A., Canetti, R., Song, D.: Efficient and secure source authentication for multicast. Proceedings of the Internet Society Network and Distributed System Security Symposium pp. 35–46 (2001)
14. Perrig, A., Canetti, R., Tygar, J.D., Song, D.: Efficient Authentication and Signing of Multicast Streams over Lossy Channels. Proceedings of the IEEE Symposium on Security and Privacy 28913, 56–73 (2000)
15. Perrig, A., Canetti, R., Tygar, J., Song, D.: The TESLA broadcast authentication protocol. CryptoBytes Summer/Fall, 2–13 (2002)
16. Perrig, A., Song, D., Canetti, R., Tygar, J., Briscoe, B.: Timed efficient stream loss-tolerant authentication (TESLA): Multicast source authentication transform introduction. The Internet Society RFC:4082, 1–22 (2005)
17. Perrig, A., Szewczyk, R., Tygar, J., Wen, V., Culler, D.E.: Spins: Security protocols for sensor networks. Wireless networks 8(5), 521–534 (2002)
18. Reyzin, L., Reyzin, N.: Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying. Information Security and Privacy 2384, 1–47 (2002)
19. Studer, A., Bai, F., Bellur, B., Perrig, A.: Flexible, extensible, and efficient VANET authentication. Journal of Communications and Networks 11, 574–588 (2009)
20. Tuffner, F.: Phasor Measurement Unit Application Data Requirements. Tech. rep., Pacific Northwest National Laboratory (2014)
21. UCTE: Final Report of the Investigation Committee on the 28 September 2003 Blackout in Italy. Tech. Rep. April, Union for the Coordination of the Transmission of Electricity (2004)
22. Ugus, O., Westhoff, D., Bohli, J.M.: A rom-friendly secure code update mechanism for wsns using a stateful-verifier τ-time signature scheme. In: Proceedings of the second ACM conference on Wireless network security. pp. 29–40. ACM (2009)
23. Wang, Q., Khurana, H., Huang, Y., Nahrstedt, K.: Time valid one-time signature for time-critical multicast data authentication. Proceedings - IEEE INFOCOM pp. 1233–1241 (2009)
24. Wang, W., Lu, Z.: Cyber security in the Smart Grid: Survey and challenges. Computer Networks 57(5), 1344–1371 (April 2013)
25. Zhu, K., Nordstrom, L., Al-Hammouri, A.: Examination of data delay and packet loss for wide-area monitoring and control systems. In: Energy Conference and Exhibition (ENERGYCON), 2012 IEEE International. pp. 927–934 (Sept 2012)