# Experimental Comparison of Multicast Authentication for Wide Area Monitoring Systems

Teklemariam Tsegay Tesfay, *Student Member, IEEE,* and Jean-Yves Le Boudec, *Fellow, IEEE*

***Abstract*—Multicast is proposed as a preferred communication mechanism for many power grid applications. One of the biggest challenges for multicast in smart grid is ensuring source authentication without violating the stringent time requirement. The research community and standardization bodies have proposed several authentication mechanisms for smart grid multicast applications. In this paper, we evaluate different authentication schemes and identify the best candidates for phasor data communication in wide area monitoring systems (WAMS). We first do an extensive literature review of existing solutions and establish a short list of schemes to evaluate. Second we make an experimental comparison of the chosen schemes in an operational smart grid pilot and evaluate the performance of these schemes by using the following metrics: computation, communication and key management overheads. The best candidates we consider are two variants of ECDSA, TV-HORS and three variants of Incomplete-key-set. We find ECDSA without pre-computed tokens and all the Incomplete-key-set variants are inapplicable for WAMS due to their high computation overhead. The ECDSA variant that uses pre-computed tokens and TV-HORS perform well in all metrics; however, TV-HORS has potential drawbacks due to a large key management overhead as a result of the frequent distribution of a large public key per source.**

***Index Terms*—Multicast source authentication, Smart grid security, Wide area monitoring systems.**

## I. Introduction

The smart grid, a superimposition of cyber infrastructure on a physical power system infrastructure, is envisioned to provide a reliable and efficient power supply with a smooth integration of renewables. Smart grid is a generic term that comprises different systems. Advanced metering systems, demand response management systems, substation automation systems, and wide area monitoring systems (WAMS) are a few of several systems that define a smart grid.

The cyber infrastructure in a smart grid facilitates two-way communication of sensing (metering) data and control signals among field devices and control centres. The field devices and the communication infrastructure usually span a large unprotected geographic area. One challenge for such a system is protecting against cyber attacks; in particular, guaranteeing message source authenticity to data consumers is difficult. Different systems in a smart grid use different communication paradigms, have different real-time requirements and the devices they use have different levels of resource constraints. Hence, there is no a one-size-fits-all security solution that works for all systems.

In this paper, we focus on identifying the best source authentication scheme for phasor data communication in wide area monitoring systems (WAMS). WAMS use high-resolution phasor data from several phasor measurement units (PMUs) to provide real-time information about a power grid's state and can be used to trigger corrective actions to maintain reliability. The North American Synchrophasor Initiative (NASPI) was founded to facilitate the deployment and use of synchrophasor technology for grid reliability and efficiency [1]. Although a few years ago there were only a few hundred PMUs deployed across the North American power grid and elsewhere, their adoption has exponentially increased due to their perceived benefits and their reduced cost [2]. PMUs provide time-synchronized data at a high data rate compared to supervisory control and data acquisition (SCADA) systems, typically from 30 to 60 samples/second. This enables power grid operators to have real-time situational awareness of their grid, which in turn enables them to implement fast response to unstable conditions observed in the grid. Depending on the nature of the different control applications that use WAMS [2], the overall delay budget for synchrophasor data ranges from 4 to 20 ms [3]. Most of this budget is consumed by the communication and computation excluding security related operations. Therefore, the additional delay due to security is preferred to be in the order of sub-milliseconds.

IP multicast is envisioned to be a preferred communication paradigm for PMU to Phasor Data Concentrator (PDC) streaming [4], [5] because it is efficient for one-to-many communication in that it relieves a PMU from sending several copies of the same packet destined to multiple PDCs. Besides, since a multicast group address is used as a destination, new receivers can be added to an already operational WAMS seamlessly without any setting changes to other PMUs or other PDCs in the group. Multiple receivers are used for different reasons. A common reason is to support redundant PDCs for reliability. Another common reason is to have the SCADA system and archive servers receive the PMU data for supervision, fault detection and post mortem analysis. In some cases, a utility shares synchrophasor data with other neighboring organizations so that all utilities have a common understanding of the state of the entire grid, which allows them to better respond to detected conditions across the grid.

In spite of its benefits, multicast also comes with its own security challenges. More specifically, designing a multicast source authentication scheme for time-critical systems such as WAMS is a challenging problem [6], [7]. As a result, this

T.T. Tesfay and J.-Y. Le Boudec are with the Laboratory for Communications and Applications 2, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015, Lausanne, Switzerland.

This work was supported by the Nano-Tera Swiss National Science Foundation Project S3-Grids.

problem is extensively studied by the research community [8]. Guaranteeing source authentication (thereby message integrity) is crucial for WAMS because any tampering of the synchrophasor data while in transit or injection of bogus data by an attacker leads to wrong real-time situational awareness of the grid; which in turn can lead to issuing wrong corrective measures with catastrophic consequences.

A trivial approach to providing multicast source authentication is to use a shared key (group key) scheme that uses message authentication codes (MACs). Several studies have proposed this as fast authentication mechanism for different smart grid applications [9], [10]. Although such a scheme is computationally fast and provides group authentication, it does not give any protection against an untrusted receiver since such a receiver can impersonate the source using the shared key. Group authentication can be considered sufficient for homogenous substation automation systems as we can assume that if one of the receivers in such a system is compromised, other receivers are also likely to be compromised. In this paper we consider WAMS, which are heterogeneous systems compared to substation automation systems. In WAMS, receivers are not necessarily colocated and may not have the same level of security. Therefore, group key based authentication is not viable in our framework because an attacker needs to compromise only one receiver or source to compromise the whole network.

An efficient multicast authentication requires a source of asymmetry in the authentication information. In other words, receivers should be able to verify the authentication information, but should not be able to generate valid authentication information [7], [8]. Different schemes use different sources of asymmetry. Some schemes use as a source of asymmetry the difference in the number of symmetric key materials that sources and receivers know [11]; others use time [12] and yet others use the computational intractability of the cryptographic primitives used to generate the keys (e.g., one-wayness of a function, collusion resistance of hash functions, factoring difficulty, discrete log problem) [13]–[15].

In this paper, we evaluate the different multicast authentication schemes that use asymmetry in the authentication information and identify the best candidate for WAMS. The set of metrics we use to evaluate the performance of these schemes are computation overhead, communication overhead and key management (key generation, distribution and storage) overhead. From the literature review, the short-list we identify for further evaluation are two variants of elliptic curve digital signature algorithm (ECDSA) [14], "time valid hash to obtain random subsets" (TV-HORS) [13] and three variants of Incomplete-key-set [11]. An experimental comparison of the short-list is then made in an operational wide area monitoring system that deploys the National Instruments CompactRIO 9068 based PMUs and phasor data concentrators (PDCs) to monitor a medium-voltage distribution network on the EPFL campus [16]. To the best of our knowledge, we are the first to perform an experimental comparison of different authentication schemes using actual PMUs deployed on an operational WAMS.

From our experiment, we find that even though the Incomplete-key-set variants use only symmetric key operations, their high computation and communication overheads make them impractical for WAMS based real-time applications. The ECDSA with no pre-computed tokens has low communication and key management overheads; however it has high computation overhead due to a slow key generation at resource-constrained PMUs. Therefore, for all practical purposes it requires hardware support in PMUs. The ECDSA variant which uses pre-computed tokens for fast signature generation has small computation and communication overheads which make it an ideal candidate for WAMS. TV-HORS also has low computation and communication overheads, but it has a large key management overhead as it requires frequent distribution of a large public key that needs to be reliably delivered to each receiver within a specified time window.

The rest of the paper is organized as follows. In Section II we present the state of the art. In Section III, we provide an in-depth discussion of the short-listed schemes. We describe the wide area monitoring system for the EPFL active distribution network which we use as our testbed for the experimental comparison of the schemes in Section IV. We present the experimental results and comparison of the schemes in Section V. Finally, in Section VI we conclude the paper.

## II. Authentication Mechanisms for IP Multicast

In this section, we cover the state of the art for multicast authentication. We also identify which source authentication schemes are more feasible for phasor data communication in wide area monitoring systems (WAMS).

### A. Asymmetric cryptography based schemes

Authentication schemes in this category include all schemes that are based on digital signatures, such as RSA and ECDSA [17]. Sources use their private keys to sign messages and receivers use the source's public key to verify received message source authenticity. These schemes are scalable in that they require a single small-size public/private key pair for every multicast source. However, directly applying these schemes for most real-time (e.g., smart grid) applications is a challenge because of their expensive computation overhead. The IEC standardization body in its IEC 62351- 6 [18] standard suggests that RSA be used to authenticate IEC 61850 Generic Object Oriented Substation Event (GOOSE) / Sampled Measured Values (SMV) messages that have a 4ms response time. However, resource constrained intelligent electronic devices (IEDs) in substations are generally incapable of computing and verifying a digital signature using the RSA algorithm within the required response time. Yavuz in [19] proposed a fast RSA based scheme by exploiting an existing structure in command and control messages. Such a scheme, though efficient, is not applicable for WAMS because the structure assumed in [19] is not present in PMU measurements. Hohlbaum et al. [20] show that, with today's IED's hardware, the software implementation of digital signatures would not meet the real-time requirements of GOOSE/SMV messages. They also show the FPGA implementation of RSA signature with a key length of 1024 bits is not feasible for systems that have less than 4ms response time requirement. However, an RSA implementation

on hardwares like ASIC platforms and specialized crypto-chips are shown to be feasible solutions.

The cost of specialized hardware are expected to be affordable in the future that we can imagine digital signature solutions be preferred solutions in future smart grid devices. Therefore, we consider digital signature based solutions as one of the candidates for multicast authentication. More specifically, we choose ECDSA as the preferred candidate among digital signature schemes to be included in the short-list, as it has a shorter public/private key length and signature size compared to RSA for a similar security level.

### B. One-time signature (OTS) schemes

One-time signature were first proposed by Lamport [21] and by Rabin [22]. Subsequent works on OTS [13], [23], [24] improved the signature length and computation overhead required for signing and verification. Law et al. in [25] provide a simulation-validated mathematical analysis of the different OTS schemes and identify TV- HORS [13] as the favourable authentication scheme for real-time applications in terms of providing a balanced computation and communication efficiencies relative to security level. In a different context from WAMS, Lu et. al in [26] compare by simulation TV-HORS with RSA when applied for multicast authentication in substation automation systems. Their results show that TV-HORS performs better than RSA, in terms of computation cost. From our literature review and from works that did theoretical and simulated comparison of OTS systems, TV-HORS is shown to be the preferred scheme among OTS schemes. Therefore, TV-HORS is included in our short-list of candidate schemes for further evaluation.

### C. Message authentication code (MAC) based schemes

MAC based schemes use a shared symmetric key between a sender and a receiver to generate a cryptographically secure authentication tag for a given message The simplest scheme in this category uses a group key shared among the multicast source and all the receivers. For example, a multicast extension to IPsec (RFC 5374) uses group keys to provide message authenticity and confidentiality. Secure distribution of the key to the multicast group members is handled by the group domain of interpretation protocol (GDOI, RFC 6407). The IEC 61850-90-5 [9] standard specifies the multicast extension of IPsec to secure synchrophasor data. Zhang and Gunter [10] also propose using IPsec for securing multicast data in substation automation and show the stringent latency constraints (less than 4ms) can be satisfied with their solution. The problem with all group key based solutions is they do not provide protection against a malicious receiver, i.e., any receiver that has the shared key can impersonate a legitimate source.

Another variant of the symmetric key based solution uses a secret-information asymmetry to cope with the impersonation problem stated above. Canetti et al. [11] propose such a scalable scheme suitable for systems with a large number of multicast receivers. In this scheme, the source knows a set of secret keys to authenticate a multicast message and each receiver knows only a subset of these keys that enable it only to verify the authenticity of received messages without being able to generate valid authentication information for messages [8]. The source attaches MACs computed using all its keys to the messages and each receiver uses its subset of keys to verify the authenticity of the received message. We refer to this scheme as the *Incomplete-key-set* scheme [26].

As the Incomplete-key-set scheme uses only fast MAC computations and does not require buffering before authentication, we include this scheme in the short-list of candidate schemes for further evaluation. In Section III-C, we provide a more detailed description of the scheme.

### D. Delayed key disclosure schemes

Like the schemes in II-C, schemes in this category use a keyed-hash message authentication code (HMAC) for source and message authentication. The main difference between the two categories is the source of asymmetry, i.e. delayed key disclosure based schemes use time as a source of asymmetry. The source computes the HMAC of a message by using a symmetric key that only it knows. The receiver buffers the message until it receives the authentication key from the source. The source then discloses the key in its subsequent messages. Timed efficient stream loss-tolerant authentication (TESLA) [12] and its variants [27], [28] are examples of this scheme. To minimize the effect of packet losses, TESLA employs a chain of authentication keys linked to each other by a pseudo random function. Each key in the key chain is the image of the next key under the pseudo random function.

Delayed key disclosure schemes have low computation overhead (only one MAC function) and low communication overhead. The drawback with these schemes is they need to buffer messages, which makes them inapplicable for real-time smart grid applications like WAMS. Thus, we do not include schemes from this category in our short-list.

### E. Signature amortization schemes

Signature amortization refers to using a single signature for authenticating a group of multicast packets, thereby spreading (amortizing) the signature verification cost across this group of packets [29]. A receiver has to assemble all the packets in the group before verifying their collective signature. As the introduced delay due to buffering makes them inapplicable for real-time applications, we do not consider schemes in this category for further evaluations.

Table I provides a summary of the different authentication schemes with respect to some desirable properties for WAMS. We have selected these desirable properties that are applicable for WAMS from those identified in [7] and [19]. A perfect scheme would be one that performs well in all the identified properties. As can be seen from the table none of the schemes satisfy that requirement. The subset of schemes we have chosen for further evaluation are those that satisfy the first three properties.

## III. Candidate multicast authentication schemes for wide area monitoring systems

In this section, we give a description of the three multicast authentication schemes that we identified in Section II as candidates for wide area monitoring systems.

TABLE I: Summary of different multicast authentication schemes with respect to different desirable properties for WAMS.

| | PKC | | OTS | MAC based | | Delayed disclosure | Amortized |
|---|---|---|---|---|---|---|---|
| | RSA | ECDSA | TV-HORS | Group key | IKS | TESLA | RSA based |
| Immediate authentication (no buffering) | Yes | Yes | Yes | Yes | Yes | No | No |
| Provides asymmetry | Yes | Yes | Yes | No | Yes | Yes | Yes |
| Robust to data packet loss | Yes | Yes | Yes | Yes | Yes | Partial | Partial |
| Scalable for large systems | Yes | Yes | Moderate | Yes | No | Yes | Yes |
| Free from time-bounded security | Yes | Yes | No | Yes | Yes | No | Yes |
| Low computation overhead | No | No | Yes | Yes | No | Yes | No |
| Low communication overhead | Yes | Yes | Yes | Yes | No | Yes | Yes |
| Low key storage at source | Yes | Yes | No | Yes | No | Moderate | Yes |
| Low key storage at receiver | Yes | Yes | No | Yes | No | Yes | Yes |

IKS: Independent-key-set; PKC: Public key cryptography

### A. Elliptic Curve Digital Signature Algorithm (ECDSA)

The elliptic curve digital signature algorithm (ECDSA) is a public-key authentication scheme whose security is based on the computational intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP) [14]. ECDSA provides the same level of security as other digital signatures, such as RSA, but with a smaller key size. Smaller keys enable ECDSA to have a faster computation time. For this reason, ECDSA is the digital signature scheme of choice for new applications: for example, Bitcoin relies on ECDSA for its security.

Below, we provide a brief description of the steps required to set up an ECDSA based multicast authentication system. More specifically we describe the domain parameter setup, key pair generation, signature generation and signature verification.

*1) Domain parameters setup:* The public/private key pairs used by ECDSA are generated with respect to a particular set of domain parameters $(p, a, b, G, n)$, where $p$ is the prime modulus, $a$ and $b$ are coefficients of the elliptic curve, $G$ is a group generator of prime order $n$. For better security, the elliptic curve should be chosen from a small set of elliptic curves referenced as NIST Recommended Elliptic Curves in FIPS publication 186 [17].

*2) Key pair generation:* Once the domain parameters are chosen, public/private key pair is generated as follows:

(a) Private key is a random integer $d \in [1, n-1]$.
(b) Public key $Q = dG$ is a point on the elliptic curve.

*3) Signature generation:* Given a hash function $h$ and a sender's key pair $(d, Q)$, a message $m$ is signed as follows:

(a) Select random $k \in [1, n-1]$.
(b) Compute $(x_1, y_1) = kG$.
(c) $r = x_1 \bmod n$. If $r = 0$, go back to step $a$.
(d) Compute $s = k^{-1}(h(m) + rd) \bmod n$.
If $s = 0$, go back to step $a$.
(e) The signature for message $m$ is the pair $(s, r)$.

*4) Signature verification:* Give a sender's public key $Q$, the authenticity of a received message $m$ is verified as follows:

(a) Compute $(x_2, y_2) = s^{-1}(h(m)G + rQ)$.
(b) Verification succeeds if $x_2 \equiv r \bmod n$ and $r,s \in [1, n-1]$.

An interesting feature of ECDSA is that signature generation is faster than signature verification. This is a desirable feature for applications like WAMS because message sources (PMUs) are more resource constrained than message receivers (PDCs). Even with such asymmetry, signature generation is still expensive. A typical approach to achieve fast signature generation is to pre-compute $r$ and $k$'s modular inverse $k^{-1}$ before the message is known [30]. By pre-computing $\aleph$ of these tokens offline, we later use them to sign $\aleph$ messages as they appear at a minimum cost. In this paper, we evaluate the performance of ECDSA signature generation with and without pre-computed tokens.

### B. Time Valid Hash to Obtain Random Subsets (TV-HORS)

TV-HORS [13] is an extension of hash to obtain random subsets (HORS) [23] authentication scheme. TV-HORS inherits HORS's advantages of fast message signing and verification. TV-HORS achieves small signature size and faster computational efficiency by signing only part of the hash of the message and by using a time-bounded signatures to prevent signature forgery. The signature period (a.k.a., epoch) is the maximum possible duration a signature can be exposed before it is verified. This duration has to be short enough so that an attacker cannot get a partial-hash collision of the signed message within that time duration.

One drawback of TV-HORS is the need for a periodic exchange of a large public key. TV-HORS uses two approaches to decrease the public key refresh rate: (1) It reuses its private key to sign multiple messages within a given epoch, i.e., it functions as a multiple-time instead of a one-time signature scheme. (2) It uses multiple key pairs linked together by using one-way hash chains, as show in Figure 1, to authenticate a large number of streaming packets without needing to redistribute a new public key at the end of every epoch.

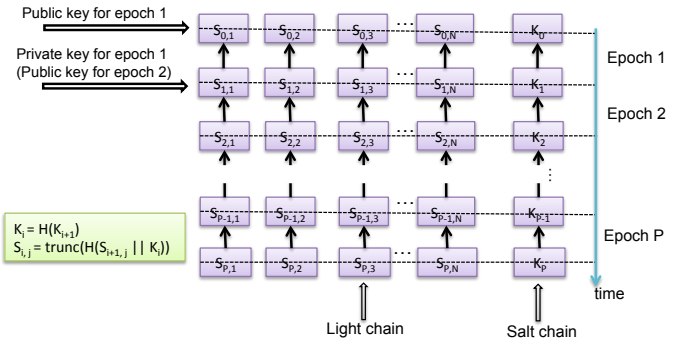



Fig. 1: TV-HORS key pairs linked using one-way hash chains. At epoch $j$, the light chain $s(j, \_)$ and the salt $k_j$ form the active private key. This private key can sign upto $v$ message within that epoch. A session has a total of $P$ epochs. A key chain is refreshed at the end of epoch $P$.

Though the "multiple timed-ness" feature improves the public key refresh rate, it also has security ramifications. It exposes more elements in the private key with every signed

message. Thus, it provides an attacker with more opportunities to forge a message using the released private key elements.

The security level $L$ for TV-HORS is expressed as a function of three parameters: the maximum number of messages that can be signed by a private key within an epoch $v$, the number of elements in a private key $N$ and the number of elements in a signature $t$. As shown in [13], $L = t\log_2(N/vt)$. The security level $L$ is a security parameter such that an adversary has to compute $2^L$ hash computations on average to obtain a valid signature for a new message. Hence the TV-HORS parameters $N, t, v$ should be chosen such that the above formula satisfies a required security level $L$.

### C. Incomplete-key-set

The basic idea behind the Incomplete-key-set scheme is the sender appends to each multicast message multiple MACs computed by using different symmetric keys. The asymmetry between senders and receivers is provided by the fact that the source knows more secret keys than each receiver.

Below we present three variants of this scheme that apply for two different scenarios.

*1) Incomplete-key-set for a small number of receivers per group:* In WAMS where the number of receivers is small (in the order of tens), implementing a variant that we refer to as *perfectly-secure Incomplete-key-set* is sufficient. For a multicast group of $R$ receivers and any number of sources, this scheme uses a total of $R$ primary secret keys $\kappa = \{k_1, ..., k_R\}$ from which $R$ secondary secret keys $\kappa_s = \{f(s, k_1), ..., f(s, k_R)\}$ are generated and assigned to each source $s$, where $f(.)$ is a pseudo-random function. Each receiver $r$ is assigned a distinct primary key $k_r$ from the set $\kappa$. The source authenticates a message $m$ by computing $R$ MACs using its $R$ secondary secrets and concatenates all the MACs with the message. Each receiver $r$ computes the secondary key of $s$ that corresponds to its primary key $k_r$ and verifies the authenticity of the message by verifying the MAC that was computed using this secondary key. However, it is not a scalable solution since the communication overhead (size of the MACs) grows linearly with the number of receivers.

*2) Incomplete-key-set for a large number of receivers per group:* In a system where there are a large number of multicast receivers, Canetti et al. [11] proposed a scheme that we will refer to as the *basic Incomplete-key-set* scheme. This addresses the scalability issue associated with the variant introduced above. This scheme uses a set of $l < R$ primary keys $\kappa = \{k_1, ..., k_l\}$ from which a set of $l$ secondary keys $\kappa_s = \{f(s, k_1), ..., f(s, k_l)\}$ are assigned to each multicast source $s$. Each receiver $r$ is assigned a set $\kappa_r$ of primary keys such that $\kappa_r \subset \kappa$. When sender $s$ wants to multicast message $m$, it computes $l$ MACs using the secondary keys in $\kappa_s$ and sends the message $m$, along with the $l$ MACs. On receiving a message from sender $s$, receiver $r$ computes the secondary keys of $s$ with the primary keys in $\kappa_r$. It then verifies all the MACs that were computed using these secondary keys. If any of these MACs is incorrect, then $r$ rejects the message.

The basic Incomplete-key-set scheme is susceptible to collusion attacks. A group of fraudulent receivers can collude among each other such that for each receiver $j$ in the fraudulent group, $\bigcup \kappa_j$ can completely cover the key subset $\kappa_u$ of a given receiver $u$ with a certain probability.

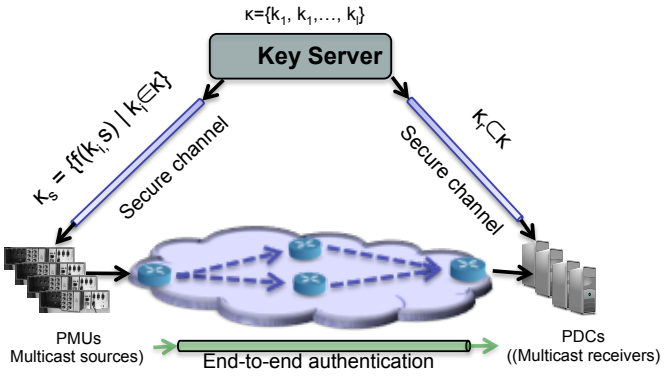



Fig. 2: Key distribution for the Incomplete-key-set authentication scheme.

Let a multicast group have a maximum number of $w$ corrupt users and let $q$ be the probability that $\kappa_u$ for any receiver $u$ is completely covered by the subsets held by the coalition members. The authors in [11] show that the number of primary keys $l$ is given by $l = e(w+1)ln(1/q)$. Each receiver $r$ obtains a subset $\kappa_r$ of primary keys such that $|\kappa_r| = e.ln(1/q)$.

Depending on the values of the system parameter $w$ and $q$, the number of keys $l$ can be large thus the communication overhead can be large. The authors in [11] propose a *communication-efficient* variant of the basic scheme that uses MACs with a single bit as output so that the authentication information is reduced to only $l$ bits. For such a setting, the number of MAC computations are four times that of the basic scheme, i.e., the total number of primary keys $l$ and $|\kappa_r|$ are four times that of the basic scheme.

## IV. System Setup and Evaluation Methodology

In this section, we describe the active power distribution network that we used as a testbed to perform our experiment to compare the three multicast authentication schemes introduced in the previous section. We also introduce the performance metrics we use to evaluate the schemes.

### A. EPFL Campus Smart Grid Monitoring System

We carry out the experimental comparison of the authentication schemes on the smart grid infrastructure deployed at EPFL to monitor the power distribution network of the campus.

Figure 3 depicts the map of the EPFL campus smart grid infrastructure. The smart grid infrastructure deploys PMUs at different locations on the campus. The PMUs measure synchrphasor data at the different locations at a rate of 50 samples/second, encapsulate the data according to the IEEE C37.118.2-2011 standard [31] and multicast it over UDP to aggregation points called phasor data concentrators (PDCs). Each synchrophasor measurement from a PMU is 74 bytes long. A PDC time-aligns the measurements from the different PMUs and feeds the time-aligned synchrphasor data to a real-time state estimator that is co-located with each PDC. The output of the real-time state estimator enables us to monitor the current state of the grid. Our monitoring infrastructure of

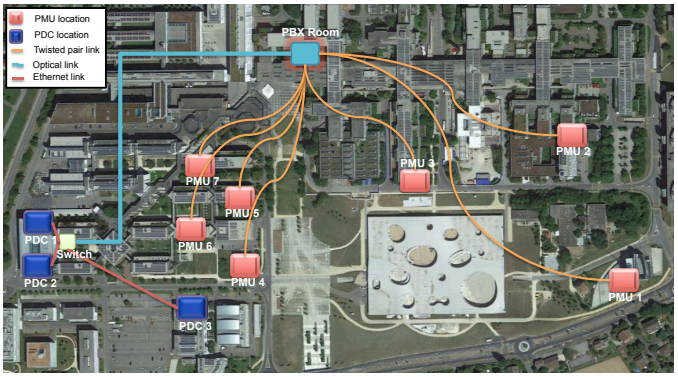



Fig. 3: The EPFL smart grid infrastracture with 7 PMUs as multicast sources and 3 PDCs as multicast receivers.

the smart grid pilot on the EPFL campus has a total of 7 PMUs and 3 PDCs in total. A more complete description of the smart grid infrastructure can be found in [16].

With no security (authentication or encryption) deployed, the overall latency between the time the synchrophasor data is sent from the PMU to the time the state estimator output is computed has a mean value of 17 ms. This relatively low latency in computing the state of the grid enables us to have a real-time grid monitoring system, which in turn enables us to implement real-time corrective measures when the state estimator output indicates a deviation from the grid's stable state. Any tampering of the synchrophasor data by an attacker, while in transit from the PMUs to the PDCs, leads to a wrong state estimator output; which in turn can lead to issuing wrong corrective measures with catastrophic consequences - thus the need for message authentication.

### B. Comparison Metrics

The set of metrics we use to compare the performance of the multicast authentication schemes are *computation overhead* per message, *communication overhead* per message and *key management overhead*. Computation overhead refers to the processing time required to generate an authentication code (signature) at the sender and to verify the authenticity of the message at the receiver. Some of the schemes we evaluate have asymmetric computation overhead for authentication and verification. An authentication scheme is considered efficient for a real-time application if the sum of the authentication and the verification time is small. Communication overhead as a metric refers to the length of the authentication data that a scheme generates per message. This metric is important especially in systems where the network bandwidth is a constraint. The third metric, key management overhead, is the cost associated with the generation, distribution and storage of the key material. The key generation overhead is the CPU time required by a PMU to generate the keys. The distribution overhead is the bandwidth required to distribute the key material to the communicating partners. The storage overhead is the amount of memory required to store the key materials.

An ideal authentication scheme for WAMS is one that has low overhead in all the metrics. However, finding a scheme that satisfies all such requirements is difficult. WAMS are real-time applications. Thus, a small computation overhead is considered a critical requirement. In contrast, utilities are likely to have dedicated state-of-the-art communication infrastructure for their sychrophasor data communication. Therefore, low communication overhead can be considered a soft requirement. The key management overhead, however, is a combination of both computation and communication overheads. Thus, a low key management overhead is also a critical requirement.

It is important to mention here that the three schemes are immune to packet losses if the packets contain application data (not key materials). For these reasons, we don't make any comparison among the schemes based on resistance to loss of packets containing application data. In contrast, packet losses during key distribution may affect the performance of a scheme and is discussed in Section V-B.

## V. Performance Evaluation and Comparisons

### A. Implementation and Parameter Settings

The multicast sources at the EPFL smart grid pilot are National Instrument's CompactRIO 9068 based PMUs with a 667 MHz dual-core ARM Cortex-A9 processor, 512 MB DDR3 memory and 1 GB nonvolatile storage running NI Linux Real-Time OS. Likewise, each receiver is a PC with an Intel 2.8 GHz Core $i7$ processor and a 4GB RAM running Ubuntu 12.04 with Linux 3.2. The source and receiver are implemented in $C$ and use OpenSSL [**?**] open source tool kit to implement the authentication schemes. We use SHA-256 whenever we need a hash output for any of the schemes.

*1) Threat model:* The attacker is assumed to have an indepth knowledge of the the power system model so that he can launch an attack similar to the one proposed by Liu et al in [32] by corrupting measurement data from a selected set of PMUs to stealthily introduce arbitrary errors in the state estimator's output of certain state variables without triggering an alarm from a bad data detection algorithm. The first ever cyber-attack on three Ukrainian regional electric power distribution companies that caused a widespread power-outage in Ukraine on December 23, 2015 demonstrates the practical feasibility of mounting such an attack successfully [33]. Moreover, we assume that an attacker has continuous remote or physical access to the communication network of the WAMS from which he can intercept and capture measurement data from the selected PMUs. We also assume the attacker has access to a cloud computing resource that is equivalent to the computing capacity of a few thousand PCs. The attacker uses the computing resources to recover the secret (private) keys used to authenticate the synchrophasor messages in real time and uses them to authenticate forged messages and send them to the receivers as if they were sent from the legitimate PMUs whose keys are compromised. Since the PMUs refresh their keys periodically, the attacker can use a compromised key only until it is refreshed. Hence, the attacker needs to continuously follow the key refresh by the PMUs and re-do the key retrieval from captured messages after every refresh.

*2) Security level and key refresh rate:* The different authentication schemes have different parameters whose values affect the schemes' performance and security level. In order to make a fair comparison of the schemes, we set their parameters so that they all have equivalent security levels. According to

[34], an ECDSA in a subgroup of $m$-bit size has an equivalent security level with a symmetric key based scheme of $m/2$ bits key-length. The security level of a symmetric key-based scheme is equal to the key length. As stated in Section III-B, the security level for TV-HORS is defined by $L$=$t\log_2(N/vt)$.

Message authentication in WAMS is a short-term issue, i.e., it is enough to guarantee that the signing key is hard to break between the signing time and the signature delivering time [35]. Therefore, in our implementation, we use short-term keys by putting a bound on the life time of these keys.

As shown in [13], it takes 16x$10^3$ workstations to break TV-HORS with $L$=54 in 6 days. Eberle et al. in [36] show it takes 3.01x$10^7$ machines equipped with ECC-processor to work together for about 24 hours to break an 112-bit ECC key ($L$=56) and 1.02x$10^{15}$ machines to break a 160-bit ECC key ($L$=80). In our experiment we considered two security levels: an intermediate security level $L$=56 and a stronger, future proof security level $L$=80. Based on the above data, we believe that a security level of $L$=56 is strong enough in the presence of an attacker with a computing capacity stated above if the keys are refreshed with in a few tens of seconds or even minutes. We have considered $L$=80, to see how the schemes compare when an attacker is likely to have more powerful computing capability in the future as cloud computing resources become more affordable.

For the intermediate security level, we generate the ECDSA key pairs from the elliptic curve domain *secp112r2* - a SECG curve over a 112-bit prime field. ECDSA keys generated from this curve have a security level $L$=56. For the Incomplete-key-set variants, we use a symmetric key-length of 56-bits. We set the TV-HORS parameters ($N$=1024, $t$=13, $v$=4), which give us $L$=56. For the stronger security level $L$=80, we use the 160-bit elliptic curve *secp160r2* for ECDSA, a symmetric key-length of 80-bits for the Incomplete-key-set and the parameters ($N$=1024, $t$=16, $v$=2) for TV-HORS. From the contour lines in Figure 4, we see that there are a range of values for $v$ and $t$ for a fixed value $N$ to achieve a required security level $L$. A contour line in the $v-t$ plane show all the possible $(v,t)$ pairs (only integer pairs) that give a value on the $L$ axis that has the same color as the contour line. We took two representative set of values for $t$ and $v$ (one for L=56 and another for $L=80$) to conduct our experiment.

For all the schemes, we use a session duration $T_s$=20 sec. The message sending rate of the PMUs in our WAMS is $\lambda$=50msgs/sec, where each message is 74 bytes long synchrophasor data. Therefore, the PMUs stream 1000 messages during one session. We assume the key material for the entire session for all the schemes are pre-generated. For TV-HORS, the key-chain length (number of epochs $P$) is given by $P$=$T_s * \lambda/v$. Therefore, for the case where $L$=56, the number of epochs $P$=250 and for the case $L$=80, $P$=500. Note that a larger $P$ value means a larger key generation and storage overhead. It also means the average verification time increases at the PDC.

The public keys for ECDSA and for TV-HORS and the symmetric keys for the Incomplete-key-set that are used during session $i$ are pre-generated and distributed during session $i-1$. Similarly, for the ECDSA with pre-computed tokens, all the tokens required for the entire session $i$ are locally pre-computed by each PMU during session $i-1$. The public keys for TV-HORS and ECDSA are multicast to all receivers in an authenticated manner. For the Incomplete-key-set the keys are distributed from the key server to PMUs and PDC using a secure unicast channel. In our implementation, the public keys are distributed only once. However, to guarantee a reliable delivery of the keys, we suggest implementing the progressive public key distribution (PPKD) scheme proposed in [35]. Note that the relative difference in the key management overhead between ECDSA and TV-HORS remains the same even when the reliable key distribution scheme is implemented.

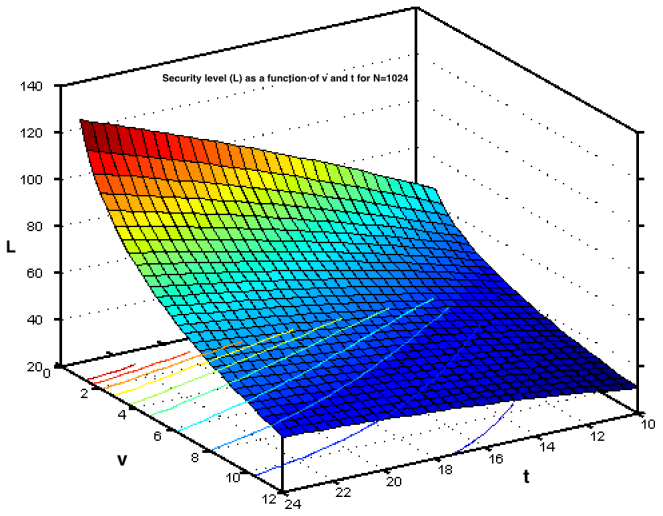



Fig. 4: TV-HORS security level (L) as a function of v and t for a fixed $N$=1024.

Following the proposals in [13], we use 48-bit light-chain elements and 80-bit salt-chain elements for TV-HORS. These parameters along with the $t$ value affect the signature length. For the perfectly-secure Incomplete-key-set we assume a total number of receivers equal to 50. For the basic and the communication-efficient variants of the Incomplete-key-set, we set the system parameters $w=10$ and $q=10^{-4}$.

### B. Performance results and comparison

In Tables II and III, we present experimental results for the performance of the candidate authentication schemes. The results show how the performance of the schemes vary depending on the values of corresponding parameters for each scheme. Below, we analyse the results for the schemes and draw conclusions on which scheme provides a better security versus performance tradeoff for WAMS.

*1) Incomplete-key-set variants:* Even though these schemes use only MAC computations, the large number of such computations introduces large computation and communication overheads per message that they are inapplicable for WAMS. Besides, the Incomplete-key-set requires a key server, which is a single point of failure, whereas EDSA and TV-HORS don't use one. Furthermore, key update for the Incomplete-key-set involves setting up a unicast encrypted channel between the key server and each of the sources and receivers, while EDSA and TV-HORS require only an authenticated multicast delivery of public keys. Therefore, given the large number of sources

TABLE II: Key management overhead of different multicast authentication schemes.

| Scheme | Key management overhead per session (20 sec) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | key generation time at PMU (ms) | | key distribution overhead at PMU (bytes) | | Key storage overhead at PMU (bytes) | | key storage overhead at PDC per PMU (bytes) | |
| | L=56 | L=80 | L=56 | L=80 | L=56 | L=80 | L=56 | L=80 |
| ECDSA without precomputed tokens | 3.367 | 5.335 | 29 | 41 | 14 | 20 | 29 | 41 |
| ECDSA with precomputed tokens | 3'340.367 | 5'447.335 | 29 | 41 | 28'014 | 40'020 | 29 | 41 |
| TV-HORS | 523.439 | 1'047.332 | 6'154 | 6'154 | 1'538'500 | 3'077'000 | 6'154 | 6'154 |
| Basic Incomplete-key-set | 0 | 0 | 1'932 | 2'760 | 1'932 | 2'760 | 175 | 250 |
| Comm. efficient Incomplete-key-set | 0 | 0 | 7'728 | 11'040 | 7'728 | 11'040 | 700 | 1'000 |
| Perfectly-secure Incomplete-key-set | 0 | 0 | 350 | 500 | 350 | 500 | 7 | 10 |

TABLE III: Performance comparison of multicast authentication schemes using per message computation and communication overheads.

| Scheme | Computation overhead per synchrophasor message | | | | | | Communication overhead (bytes) per synchrophasor message | |
|---|---|---|---|---|---|---|---|---|
| | Auth. time (ms) | | Verif. time (ms) | | Total (ms) | | | |
| | L=56 | L=80 | L=56 | L=80 | L=56 | L=80 | L=56 | L=80 |
| ECDSA without precomputed tokens | 3.431 | 5.563 | 0.223 | 0.327 | 3.654 | 5.890 | 34 | 48 |
| ECDSA with precomputed tokens | 0.104 | 0.111 | 0.223 | 0.331 | 0.327 | 0.442 | 34 | 48 |
| TV-HORS | 0.014 | 0.014 | 0.110 | 0.217 | 0.124 | 0.231 | 88 | 106 |
| Basic Incomplete-key-set | 4.559 | 4.589 | 0.068 | 0.069 | 4.627 | 4.658 | 1'932 | 2'760 |
| Comm. efficient Incomplete-key-set | 18.151 | 18.361 | 0.172 | 0.181 | 18.323 | 18.542 | 138 | 138 |
| Perfectly-secure Incomplete-key-set | 0.848 | 0.853 | 0.018 | 0.019 | 0.866 | 0.872 | 350 | 500 |

(and receivers) in WAMS, the Incomplete-key-set schemes is inefficient from the key server's point of view.

*2) ECDSA variants:* The ECDSA without pre-computed tokens scheme performs best in all metrics except in the computation overhead per message. The computation overhead for both security levels is high, which makes it unsuited for WAMS applications that have strict real-time requirement. Adding a cryptographic accelerator hardware to PMUs is one way to speed up signature generation.

Implementing ECDSA with pre-computed tokens significantly improves the computation overhead per message. The pre-computation of the tokens also introduces a non-negligible key-generation overhead (we consider token-generation part of the key generation overhead). However, the tokens for session $i$ are generated during session $i$-1. Hence a token-generation times in Table II for both security levels during a 20 second long session is within the computational capability of the kind of PMUs deployed in our smart grid. Besides, there is no significant change in the signing overhead between $L$=56 and $L$=80. The small increase in the overall computation overhead can be mitigated by deploying more powerful PDCs or by implementing an optimized ECDSA verification (which we have not implemented). Therefore, the sub-millisecond computation overheads and low communication overheads of ECDSA with pre-computed tokens for both security levels make it an ideal scheme for WAMS applications with real-time requirements for the foreseeable future. This finding is contrary to the generally accepted view that public key cryptography is inapplicable for real-time applications.

*3) TV-HORS:* TV-HORS has the lowest computation overhead and relatively low communication overhead per message. The only drawback of TV-HORS is that it requires frequently refreshing the public/private key pair and sending a large public key message to all receivers. WAMS are normally characterized by a large number of PMUs. Unless a proper randomization of key distribution is implemented, a large public key ($\approx$ 6kbytes) per PMU can cause periodic burst synchronization of packets that can have significant effect on the network bandwidth that could lead to synchrophasor packet loses. The burst of packets from each PMU can also have a non-negligible computation overhead on the receivers if the number of PMUs is in the order of hundreds or thousands. This effect is magnified if the public key has to be sent multiple times to guarantee reliable delivery.

Lu et al. in [26] identify two potential threats in TV-HORS when applied to substation automation systems (SAS) - *delay compression attack* and *key depletion attack*. The sending rate in WAMS is much slower than that of SAS - typically 50 msgs/sec; whereas a typical rate for SMV messages in SAS is 4800 msgs/sec. In our implementation a signing-key update occurs at the end of every epoch. An epoch duration of 80 ms for $L$=56 or $40ms$ for $L$=80 is long enough for any synchrophasor message to be verified within this time period. In fact, the overall end-to-end delay for phasor messages in our smart grid is less than 4 ms. Therefore, the *delay compression attack* is not an issue for WAMS. Moreover, TV-HORS replenishes its key-chain at the end of the last epoch. The time required to generate the whole key-chain for $P$=500 is only 1.047 sec (Table II). Given the relatively lower message sending rate of PMUs, pre-generating the key-chain during the 20 sec duration of session $i-1$ for session $i$ is within the computational capacity of the PMUs we used in our experiment. Hence, the *key depletion attack* (key generation speed being slower than the key consumption speed) can also be ignored as an issue in WAMS. Finally, the comparison between RSA and TV-HORS in [26] is unfair since the chosen security levels for the two schemes are not the same.

From the above observations, we can conclude ECDSA with pre-computed tokens is the preferred scheme for WAMS applications. In spite of TV-HORS' desirable low computation overhead, it has inherent drawbacks due its hard-deadline requirement to deliver a large public key to receivers within a short duration. Each private key in a TV-HORS key chain has a time window during which it can be used to sign messages. These messages must be verified by the receiver during this assigned time window or else the message is discarded by the receiver. The private key cannot be used to sign messages sent after its time window expires. By the end of the $P^{th}$ epoch, the

last private key in the key-chain will be used to sign the $v^{th}$ message of that epoch. Beyond that epoch, the multicast source has to use a new key-chain to sign new messages. However, if the public key for this new key chain is not successfully communicated to the PDCs, they will not be able to verify the messages signed using the private keys from the new key-chain. In our experiment, TV-HORS has only 20 sec to reliably deliver a large public key that is required for the next 20 sec session. As explained above, this 20 sec duration is a hard-deadline since the old key-chain cannot be used to sign more than the number of messages transmitted in 20 sec.

In contrast, ECDSA has a time window of 20 sec to deliver a relatively small public key for the next session. Besides, the 20 sec session duration for ECDSA is a conservative value. Hence, ECDSA could continue to use its old public/private key pair until the next public key is reliably delivered even beyond the 20 sec time window. The only means to extend the life time of the private/public key-chain for TV-HORS to increase $P$, which in turn introduces key generation, storage and verification overheads.

The two security levels we consider in our experiment are relatively high if we assume an attacker with low computational capabilities. Therefore, utilities who want to protect their WAMS against such an attacker may be willing to consider security levels less than 56. From the results in Table III we see that when the security level is decreased, the improvement in ECDSA's signing and verification times are much more than the other two schemes'. Hence, for lower security levels, ECDSA with pre-computed tokens is still the preferred scheme for such systems since it will still have lower overheads in all the other metrics.

### C. Support for addition and revocation

All the three schemes support dynamic addition (revocation) of senders and receivers to (from) a multicast group. In all the three schemes, we assume there is a multicast group controller similar to the one described in [10] that is responsible for granting and revoking group membership to PMUs and PDCs and for announcing the addition and revocation of members to the already existing members.

In all schemes addition/revocation of a receiver (PDC) does not cause any change in any of the existing group members. However, addition/revocation of a new source (PMU) to a group introduces some changes to existing PDCs. The group controller has to inform all PDCs (receivers) about the identity of the new PMU. Once informed about the new member, the PDCs will be able to receive the key material (public key for ECDSA and TV-HORS) from the new PMU that they can use to verify messages they will subsequently receive from it. For the Independent-key-set, the key server has to send the secondary key set $\kappa_s$ to the new PMU $s$. Performance wise, addition of a new PMU increases the aggregate verification time at the PDC. This increase per every additional PMU is proportional to the verification time in Table III.

Revocation of a PMU involves a controller informing all PDCs about the identity of the revoked PMU and each PDC removing the identity (thus the corresponding authentication key) of the revoked PMU from their list of authentic sources. Performance wise, revocation of a PMU decreases the aggregate verification times at the PDCs. Again, the decrease in the aggregate value per every revoked PMU is proportional to the verification time in Table III.

### D. Impact of the scale of WAMS

The aggregate verification time as well as the key storage requirement at the PDC is proportional to the total number of PMUs in a multicast group. Therefore, the aggregate time that a PDC spends processing (verifying the authenticity, decapsulating and aggregating) synchrophasor messages can be large if the number of PMUs in a group is very large. The IEEE C37.244 Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring [37] specifies a PDC uses a "wait timer" to wait for all messages to arrive from all PMUs before generating the aggregate data and passing it on to the state estimator. The value of the "wait timer" is user defined. Messages from all PMUs should be verified and aggregated before the timer expires. Therefore, a utility needs to determine the computational capacity of the PDC they deploy such that the aggregate processing time for all PMUs is within this limit. Our results in Table III for the verification time can be used to find the total number of PMUs that a PDC can support. Gomez-Exposito et al. in [38] propose a hierarchical multilevel state estimation framework to avoid using a single powerful central PDC that deals with aggregating synchrophasor data from a large number of PMUs. In such a paradigm, PDCs at the lowest level deal with only a small set of PMUs that are geographically closer to it and the PDCs at higher levels correlate pre-filtered data from PDCs in lower levels and possibly from other PMUs that are close to them. This way, multicast groups will have a manageable number of PMUs. The PDCs in the lower levels will be multicast sources in the multicast group for which the higher level PDCs are receivers. Hence, PDCs in the lowest level and in the intermediate levels can be both a receiver in one multicast group and a source in another multicast group.

## VI. Conclusion

In this paper, we have evaluated the performance of available multicast authentication schemes for WAMS. Contrary to the generally accepted notion that public key cryptography is impractical for real-time applications due to its high computation cost, we have shown that an ECDSA implementation that utilizes short-term keys and pre-computed tokens for signature generation provides the required performance for WAMS based real-time applications. TV-HORS is also widely treated as the scheme of choice for real-time applications in smart grid. Our findings show that even though TV-HORS has very low computation overhead even compared to ECDSA with pre-computed tokens, its potential drawbacks due to its hard-deadline requirement to reliably distribute a large public key makes it less preferable than ECDSA.

## References

[1] NASPI. [Online]. Available: https://www.naspi.org/home

[2] M. Patel, S. Aivaliotis, and E. Allen, "Real-Time Application of Synchrophasors Improving Reliability," North American Elecricity Reliability Corporation, Princeton, NJ, Tech. Rep.,, Oct. 2010.

[3] "IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation," *IEEE Std 1646-2004*, pp. 1–24, 2005.

[4] M. Seewald, "Building an architecture based on IP-Multicast for large phasor measurement unit (PMU) networks," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, Feb 2013.

[5] P. Myrda, J. Taft, and P. Donner, "Recommended Approach to a NASPInet Architecture," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, Jan 2012.

[6] D. Boneh, G. Durfee, and M. Franklin, "Lower bounds for multicast message authentication." Springer-Verlag, 2001.

[7] M. Luk, A. Perrig, and B. Whillock, "Seven Cardinal Properties of Sensor Network Broadcast Authentication," in *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks*, ser. SASN '06. New York, NY, USA: ACM, 2006.

[8] Y. Challal, H. Bettahar, and A. Bouabdallah, "A taxonomy of multicast data origin authentication: Issues and solutions," *Communications Surveys Tutorials, IEEE*, vol. 6, no. 3, Third 2004.

[9] IEC TC57, "Communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118," 2012.

[10] J. Zhang and C. Gunter, "Application-aware secure multicast for power grid communications," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, Oct 2010.

[11] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, Mar 1999, pp. 708–716 vol.2.

[12] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on*, 2000.

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

[14] S. Vanstone, " Responses to NIST's Proposal," Communications of the ACM, 35, American National Standards Institute, July 1992.

[15] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Commun. ACM*, vol. 21, no. 2, Feb. 1978.

[16] M. Pignati, M. Popovic, S. Barreto, R. Cherkaoui, G. Dario Flores, J.-Y. Le Boudec, M. Mohiuddin, M. Paolone, P. Romano, S. Sarri, T. Tesfay, D.-C. Tomozei, and L. Zanni, "Real-time state estimation of the EPFL-campus medium-voltage grid by using PMUs," in *Innovative Smart Grid Technologies Conference, 2015 IEEE Power Energy Society*, Feb 2015.

[17] NIST, "Federal Information Processing Standard 186-3 (FIPS186-4), Digital Signature Standard (DSS)," 2013.

[18] IEC TC/SC 57, "IEC 62351 - Power systems management and associated information exchange - Data and communications security," 2013.

[19] A. A. Yavuz, "An efficient real-time broadcast authentication scheme for command and control messages," *Trans. Info. For. Sec.*, Oct. 2014.

[20] F. Hohlbaum, M. Braendle, and F. Alvarez, "Cyber Security Practical considerations for implementing IEC 62351," ABB, Switzerland, 2010.

[21] L. Lamport, "Constructing Digital Signatures from a One Way Function," Technical Report, SRI-CSL-98, SRI Intl. Computer Science Laboratory, October 1979.

[22] M. O. Rabin, "Digitized Signatures and Public-key Functions as Intractable as Factorization," Cambridge, MA, USA, Tech. Rep., 1979.

[23] L. Reyzin and N. Reyzin, "Better than BiBa: Short One-time Signatures with Fast Signing and Verifying," in *In Seventh Australasian Conference on Information Security and Privacy (ACISP 2002*, 2002.

[24] Q. Li and G. Cao, "Multicast Authentication in the Smart Grid With One-Time Signature," *Smart Grid, IEEE Transactions on*, Dec 2011.

[25] Y. W. Law, Z. Gong, T. Luo, S. Marusic, and M. Palaniswami, "Comparative Study of Multicast Authentication Schemes with Application to Wide-area Measurement System," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '13. New York, NY, USA: ACM, 2013.

[26] X. Lu, W. Wang, and J. Ma, "Authentication and Integrity in the Smart Grid: An Empirical Study in Substation Automation Systems," *International Journal of Distributed Sensor Networks*, 2012.

[27] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wirel. Netw.*, Sep 2002.

[28] D. Liu and P. Ning, "Multilevel $\mu$-TESLA: Broadcast Authentication for Distributed Sensor Networks," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 4, Nov. 2004.

[29] C. Tartary, H. Wang, and S. Ling, "Authentication of digital streams," *Information Theory, IEEE Transactions on*, vol. 57, no. 9, Sept 2011.

[30] D. Naccache, D. M'RaÏhi, S. Vaudenay, and D. Raphaeli, *Advances in Cryptology - EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May, 1994 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, ch. Can D.S.A. be improved? Complexity trade-offs with the digital signature standard.

[31] "IEEE Standard for Synchrophasor Data Transfer for Power Systems," *IEEE Std C37.118.2-2011*, Dec 2011.

[32] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 13:1–13:33, Jun. 2011.

[33] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[34] N. Smart, S. Babbage, D. Catalano, C. Cid, B. de Weger, O. Dunkelman, C. Christian Gehrmann, L. Granboulan, T. Guneysu, and M. Ward, "ECRYPT II Yearly Report on Algorithms and Keysizes," European Network of Excellence in Cryptology (ECRYPT II), Sep. 2012.

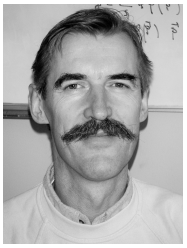
[35] R. Wang, W. Du, X. Liu, and P. Ning, "ShortPK: A Short-term Public Key Scheme for Broadcast Authentication in Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 6, no. 1, Jan 2010.

[36] H. Eberle, N. Gura, S. C. Shantz, V. Gupta, L. Rarick, and S. Sundaram, "A public-key cryptographic processor for RSA and ECC," in *Application-Specific Systems, Architectures and Processors, 2004. Proceedings. 15th IEEE International Conference on*, Sept 2004.

[37] "IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring," *IEEE Std C37.244-2013*, May 2013.

[38] A. Gomez-Exposito, A. Abur, A. de la Villa Jaen, and C. Gomez-Quiles, "A multilevel state estimation paradigm for smart grids," *Proceedings of the IEEE*, June 2011.


**Teklemariam T. Tesfay** received his B.Sc. and MSc degrees in Computer Science and Engineering in 2007 from Mekelle Institute of Technology, Ethiopia and in 2009 from the Indian Institute of Technology Bombay, India, respectively. He is currently pursuing his Ph.D. degree at the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland under the supervision of Prof. J.-Y. Le Boudec in the LCA2 lab. His research interests include identifying cybersecurity threats and proposing countermeasures for smart grid networks.



**Jean-Yves Le Boudec** is professor at EPFL and fellow of the IEEE. He graduated from Ecole Normale Supérieure de Saint-Cloud, Paris, where he obtained the Agrégation in Mathematics in 1980 and received his doctorate in 1984 from the University of Rennes, France. From 1984 to 1987 he was with INSA/IRISA, Rennes. In 1987 he joined Bell Northern Research, Ottawa, Canada, as a member of scientific staff in the Network and Product Traffic Design Department. In 1988, he joined the IBM Zurich Research Laboratory where he was manager of the Customer Premises Network Department. In 1994 he became associate professor at EPFL. His interests are in the performance and architecture of communication systems and smart grids. He co-authored a book on network calculus, which forms a foundation to many traffic control concepts in the internet, an introductory textbook on Information Sciences, and is also the author of the book "Performance Evaluation".