Comparative Study of Multicast Authentication Schemes with Application to Wide-Area Measurement System

Yee Wei Law
Department of EEE
The University of Melbourne
Australia
ywlaw@unimelb.edu.au

Zheng Gong*
School of Computer Science
South China Normal University
Guangzhou 510631, China
cis.gong@gmail.com

Tie Luo
Institute for Infocomm Research
A*STAR, Singapore
luot@i2r.a-star.edu.sg

Slaven Marusic
Department of EEE
The University of Melbourne
Australia
slaven@unimelb.edu.au

Marimuthu Palaniswami
Department of EEE
The University of Melbourne
Australia
palani@unimelb.edu.au

ABSTRACT

Multicasting refers to the transmission of a message to multiple receivers at the same time. To enable authentication of sporadic multicast messages, a conventional digital signature scheme is appropriate. To enable authentication of a multicast data stream, however, an authenticated multicast or multicast authentication (MA) scheme is necessary. An MA scheme can be constructed from a conventional digital signature scheme or a multiple-time signature (MTS) scheme. A number of MTS-based MA schemes have been proposed over the years. Here, we formally analyze four MA schemes, namely BiBa, TV-HORS, SCU+ and TSV+. Among these MA schemes, SCU+ is an MA scheme we constructed from an MTS scheme designed for secure code update, and TSV+ is our patched version of TSV, an MA scheme which we show to be vulnerable. Based on our simulation-validated analysis, which complements and at places rectifies or improves existing analyses, we compare the schemes' computational and communication efficiencies relative to their security levels. For numerical comparison of the schemes, we use parameters relevant for a smart (power) grid component called wide-area measurement system. Our comparison shows that TV-HORS, while algorithmically unsophisticated and not the best performer in all categories, is the most balanced performer. SCU+, TSV+ and by implication the schemes from which they are extended do not offer clear advantages over BiBa, the oldest among the schemes.

*Zheng Gong is also affiliated with State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China.

ASIA CCS'13, May 8–10, 2013, Hangzhou, China.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: Security and protection; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Algorithms; Security

Keywords

Multicast authentication; multiple-time signature scheme; smart grid; wide-area measurement system

1. INTRODUCTION

A **multicast authentication (MA)** scheme enables receivers of a multicast packet to authenticate the sender, and ensures no entities besides the sender can send authenticated packets to the multicast group. At the core of every MA scheme, lies a signature scheme. Conventional digital signature algorithms such as DSA and ECDSA (see FIPS 186-3) can sign a practically unlimited number of distinct messages with a private key, but they have high computation and memory requirements. The simplest digital signature-based MA scheme appends a digital signature to every multicast message, which is fine for sporadic multicast messages, but computationally prohibitive for multicast data streams. The latest research on digital signature-based MA scheme focuses on *signature amortization* [27], i.e., spreading a signature across many packets such that signature verification cost is amortized and the loss of a small number of packets does not forestall the verification of successfully received packets. However, all signature amortization schemes require a number of packets to be assembled before their collective signature can be verified, incurring a delay that can be problematic for real-time applications.

Real-time requirements have been motivating investigation of **multiple-time signature (MTS)** schemes [22] for multicast authentication. An MTS scheme can sign a fixed number of distinct messages using a public/private key pair. Although they generally produce longer signatures, they have

much lower computation and memory requirements than conventional digital signature schemes. MTS-based MA schemes are thus suitable for real-time systems, and one such system is a smart grid component called **wide-area measurement system (WAMS)**, which in fact motivates this work. A number of MTS-based MA schemes have been proposed over the years, yet due to inadequate analysis, a systematic comparison of these schemes is lacking, preventing a scientific approach to the selection of MA schemes for real-time systems, including WAMS.

Our contributions include a rigorous, simulation-validated analysis and a methodical comparison of four MTS-based MA schemes, namely BiBa [20], TV-HORS [31], SCU+ and TSV+ [13]. Among these MA schemes, SCU+ is an MA scheme we constructed from an MTS scheme designed for secure code update [29], and TSV+ [13] is a patched version of Tunable Signing and Verification (TSV) [15]. These MA and MTS schemes are chosen because they are either highly cited or tailored to smart grids. Our analysis fills the gaps of, and at places rectifies or improves existing analyses. Our comparison shows that TV-HORS has the most balanced computational and communication costs; and that contrary to common perception, recent sophisticated designs do not necessarily trump older over-criticized designs. Our work is motivated by the need to find an efficient MA scheme for the WAMS, and hence the parameters of our comparison are tailored to the WAMS. However, our analysis is system-independent, and the parameter values used for comparison can be adapted to any other system. This work is meant to serve as the first step of an ongoing series of comparative studies of MTS-based MA schemes.

This article is organized as follows. Section 2 discusses related work. Section 3 lists the mathematical symbols and notation used in this work. Section 4 presents an overview of MA schemes. We present our analysis of BiBa, TV-HORS, SCU+ and TSV+ in Section 5. In Sections 6 and 7, we introduce the WAMS and compare the four schemes using parameter values relevant for the WAMS. Section 8 concludes the paper.

2. RELATED WORK

Active research on wireless sensor networks for the past decade can be said to have spurred interest in broadcast/multicast authentication and therefore multiple-time signature (MTS) schemes. This is largely due to the resource constraints on a typical sensor node that favor low computational complexity and small code size. Recently, smart grid researchers are also turning to MTS schemes for real-time multicast authentication.

It is well known that one-time signatures (a subset of MTS schemes) were first considered by Lamport [12]. Among subsequent schemes that improve upon Lamport's impractical construction, BiBa [20] counts as a benchmark for its simple ingenuity. TESLA and its variant μTESLA [21] are more lightweight than BiBa but the use of delayed signature verification in these schemes precludes them from real-time multicast authentication. HORS [23] improves upon BiBa by generating shorter signatures for the same security level, and has inspired many variants (e.g., [14,19]) and extensions. TV-HORS [31] is an extension of HORS to an MA scheme. TSV [15] is both a variant and an extension of HORS, because it is both an MTS scheme and an MA scheme. TV-HORS and TSV were both motivated by smart grid applications, making them ideal candidates for comparison here. SCU [29] was designed for wireless sensor networks, and has an interesting design, so including it in our comparison introduces diversity. Using Katz's taxonomy [10], all schemes studied here are chain-based stateful schemes. As Steinwadt et al. [25] noted, naming a single superior MTS scheme (and accordingly, MA scheme) is nontrivial.

In the following, we discuss related work in the context of the wide-area measurement system (WAMS). In the North American SynchroPhasor Initiative (NASPI), data multicasting from a phasor measurement unit (a WAMS node) to multiple control centers is seen as a necessity. Bobba et al. [3] proposed a Policy-Based Encryption System (PBES) to secure sharing of data, including WAMS data, between utilities. PBES was not designed for securing WAMS traffic itself. NASPI has yet to standardize on an MA scheme for the purpose.

IEC 61850 is a series of standards on *substation automation*, i.e., the automation of data acquisition, control, protection, diagnostics and monitoring functions within substations (where most WAMS nodes are located). As part of the series, IEC 61850-90-5 governs the IEC 61850-compliant transmission of IEEE C37.118-formatted WAMS data. The standard specifies Group Domain of Interpretation (GDOI, see RFC 6407) for securing the distribution of group keys, and IPsec (see RFC 4301) for securing IP multicast using the group keys. However, GDOI does not support mutual authentication among group members [17, Section 4.3]. Furthermore, IPsec relies on a shared group key for encryption, which can be abused by a rogue member to forge messages to the whole group (see RFC 5374). Zhang and Gunter [33] proposed using IPsec for securing multicast WAMS data, but did not point out the pitfalls as we do here.

Researchers at the Future Renewable Electric Energy Delivery and Management (FREEDM) Systems Center implemented TV-HORS on their testbed [32], but did not provide the elaborate justification we provide here.

Recently, Law et al. [13] proposed a key management scheme for the WAMS that specifies TV-HORS for securing multicast data streams. Their choice is based on a simulation-based comparison between TV-HORS and TSV+. Our comparison here covers more schemes, and is both analytical and empirical.

Within the wireless sensor network community, several studies have been performed to evaluate the efficiency of various signature schemes. For example, Seys and Preneel [24] compared the *energy-efficiency* of ECDSA, Lamport-Diffie and HORS one-time signature schemes. Their results show that for less than 15000 signatures, HORS is the most energy-efficient, whereas for more than 15000 signatures, Lamport-Diffie is better.

3. NOTATION AND DEFINITIONS

For the ensuing discussion, the following mathematical notation is used:

H	one-way hash function;
M, c	message and counter respectively;
t	number of elements of a private key tuple;
λ	last index of a one-way chain;
$\mathscr{S}$	see Definition 1;
$\mathscr{C}_\sigma, \mathscr{C}_v$	expected number of hash operations required for signing and verification respectively;
$\mathscr{L}_\sigma$	expected number of bits of a signature;

r	number of signatures generated per epoch;
$\|x\|$	bit-length of x when x is a bit-string;
l_H	bit-length of a *truncated* hash value;
$\mathsf{PRF}(K, M)$	pseudorandom function (PRF) with key K and plaintext M;
$\mathrm{Split}_k()$	function that splits a bit-string into k sub-strings.

Note that a one-way hash function is a hash function that is preimage-resistant and second preimage-resistant. Furthermore, the following definitions are used:

DEFINITION 1. *Assuming a polynomial-time adversary $\mathcal{A}$ can successfully execute an existential forgery on a scheme $\mathcal{S}$ with probability p, then the security level of the scheme is* $\mathscr{S}(\mathcal{A}, S) \stackrel{\triangle}{=} -\log_2(p)$.

DEFINITION 2. *[26] Let f be a function from X to Y, and $x_1, \ldots, x_k \in X$. Suppose that values $y_i = f(x_i)$ have been determined for $i = 1, \ldots, k$. Then, f is k-wise independent, if for all $x \in X \setminus \{x_1, \ldots, x_k\}$ and all $y \in Y$,*

$$\Pr[H(x) = y | y_1 \leftarrow H(x_1) \wedge \cdots \wedge y_k \leftarrow H(x_k)] = 1/|Y|.$$

4. MULTICAST AUTHENTICATION USING MULTIPLE-TIME SIGNATURE SCHEMES

Most MTS schemes can be divided into the following parts:

- a private key tuple $(s_1, s_2, \ldots, s_t)$ consisting of t fixed-length random strings;
- a key generation algorithm for generating a public key tuple from a private key tuple;
- a signature generation algorithm that, based on the message to be signed, selects elements of a private key, and generate signature elements from the selected private key elements;
- a signature verification algorithm that checks if the public key elements can be derived from the received signature elements.

Constructing an MA scheme from an MTS scheme requires two key "ingredients". **The first ingredient** is *one-way chains*. Since a key pair can generate only a fixed number of signatures, to sign a message stream of unlimited length, the key pair must be refreshed once its usage limit is reached. The de facto standard technique is to use the private key of an expired key pair as the public key of a new key pair. In the case of BiBa, this means generating the one-way chain as $s_{i,\lambda}, s_{i,\lambda-1}, \ldots, s_{i,0}$, where $s_{i,j-1} = H(s_{i,j}), \forall i = 1, \ldots, t$ and $j = 1, \ldots, \lambda$. As such, $s_{i,j}$ is *both* the private key element corresponding to public key element $s_{i,j-1}$, *and* the public key element corresponding to the private key element $s_{i,j+1}$ (see Fig. 1). Since this technique requires a private key to have the same number of elements as the corresponding public key, this technique does not apply to tree-based MTS schemes (e.g., [11]). **The second ingredient** is clock/time synchronization, which is essential for the security of MTS-based MA schemes, as explained in Fig. 1. In a smart grid, this requirement is satisfiable by the draft standard IEEE PC37.238, which specifies a common profile for the use of IEEE 1588-2008 Precision Time Protocol in power system protection, control, automation and data communication applications utilizing an Ethernet communications architecture.

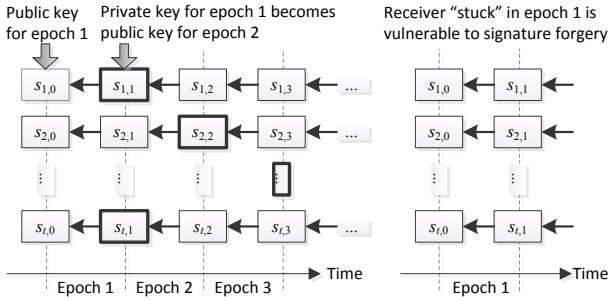

Figure 1: Some (not all) private key elements are disclosed for each signature generated. However, as time passes, an attacker can capture enough signatures (in this example, the "thick boxes") to reconstruct the whole $s_{1,1}, s_{2,1}, \ldots, s_{t,1}$. It is therefore necessary to deprecate private keys by (i) dividing time into epochs, (ii) keeping track of the active private key corresponding to the current epoch, and (iii) synchronizing the clocks of the sender and receivers.

In the approach depicted in Fig. 1, during epoch j, the active private key elements are $s_{1,j}, \ldots, s_{t,j}$. We call this approach **uniform chain traversal**. Using uniform chain traversal, in the first epoch, when a signature containing $s_{i_1,1}, s_{i_2,1}, \ldots, s_{i_k,1}$ is received, H is invoked k times to check $H(s_{i_1,1}) \stackrel{?}{=} s_{i_1,0}$, $H(s_{i_2,1}) \stackrel{?}{=} s_{i_2,0}$, and so on. Assume every private key is used to generate only one signature. In the second epoch, suppose a signature containing $s_{j_1,2}, s_{j_2,2}, \ldots, s_{j_k,2}$ is received, and $\{i_1, \ldots, i_k\} \cap \{j_1, \ldots, j_k\} = \emptyset$. This time, H would have to be invoked $2k$ times to check $H^2(s_{j_1,2}) \stackrel{?}{=} s_{j_1,0}$, $H^2(s_{j_2,2}) \stackrel{?}{=} s_{j_2,0}$, and so on. Therefore, in uniform chain traversal, the verification cost, i.e., expected number of hash operations for verifying a signature, varies from signature to signature.

An alternative approach, which we call **nonuniform chain traversal**, is meant to ensure the verification cost stays at a minimum from signature to signature. Using nonuniform chain traversal, the first active private key is $s_{1,1}, s_{2,1}, \ldots, s_{t,1}$. Without loss of generality, suppose $s_{1,1}, s_{2,1}, \ldots, s_{k,1}$ have been used for a signature, the active private key now becomes $s_{1,2}, s_{2,2}, \ldots, s_{k,2}, s_{k+1,1}, \ldots, s_{t,1}$. As explained in Fig. 1, it is essential that receivers keep track of the active private key, but the loss of synchrony between the epoch counter and the key-chain indices of the active private key means time can no longer be used as a reference. The sender can disclose the key-chain indices of the active private key with every signature, but by blocking packets to a receiver, an attacker can cause a receiver to lose track of the current active private key. Once the attacker has collected enough signatures, it will be able to forge packets to the receiver. In order to keep the verification cost at a minimum for every signature, nonuniform chain traversal inadvertently compromises the receivers' ability to track the active private key and exposes them to signature forgery. Therefore in comparison, uniform chain traversal is more robust and is adopted for all MA schemes in this paper, at the expense of higher verification cost.

The discussion above glosses over a particular caveat of one-way chains. If H is to be k-wise independent (see Definition 2), following Bradford et al.'s analysis [4], the size of

the domain of H must be $\Omega((\lambda+1)^{\lfloor k/2 \rfloor})$, where $\lambda+1$ is the length of the one-way chain. There are many ways to expand the domain of H (below, $s_{i,\lambda}$ is randomized, $i = 1, \ldots, t$, $j = 1, \ldots, \lambda$):

1. Use a separate *salt chain* $K_{j-1} = \mathsf{PRF}(K_j, 0)$, and set $s_{i,j-1} = \mathsf{PRF}(s_{i,j}, K_{i,j})$, $\forall j = 2, \ldots, \lambda$ [20].
2. Use a separate salt chain $K_{j-1} = H(K_j)$, and set $s_{i,j-1} = H(s_{i,j} \| K_j)$ [31].
3. Use a synchronized counter c_j, e.g., the epoch counter, and set $s_{i,j-1} = H(s_{i,j} \| c_j)$ [5].
4. Set $s_{i,j-1} = H(s_{i,j} \| s_{i,j+1})$, $s_{i,\lambda-1} = H(s_{i,\lambda})$ [5].

Above, note that replacing $\mathsf{PRF}(K, M)$ with $H(K \| M)$ is valid provided H can be modelled as a random oracle [2].

The performance of an MA scheme is evaluated in terms of the computational complexity of signature generation and signature verification ($\mathscr{C}_\sigma$ and $\mathscr{C}_v$), and in terms of communication overhead ($\mathscr{L}_\sigma$). However, these performance metrics are only meaningful with respect to the achievable security level ($\mathscr{S}$). In other words, we are interested in how MA schemes compare with each other in terms of (i) $\mathscr{L}_\sigma \,/\, \mathscr{S}$, (ii) $\mathscr{C}_\sigma \,/\, \mathscr{S}$, and (iii) $\mathscr{C}_v \,/\, \mathscr{S}$. The *lower* an MA scheme scores in all these metrics, the better the MA scheme. The next section presents our description and analysis of BiBa, TV-HORS, SCU+ and TSV+.

5. ANALYSIS OF MULTICAST AUTHENTICATION SCHEMES

The MA schemes BiBa, TV-HORS, SCU+ and TSV+ are analysed in terms of signing cost, verification cost, signature length, and security level.

For the assessment of security levels, the following models are used: (i) **random oracle model** [2]: hash function outputs are uniformly distributed at random; (ii) **Dolev-Yao model** [8]: an attacker can intercept, modify, fabricate messages. We add the condition that is implicit in the literature (e.g., in [20]): an attacker cannot completely disrupt clock synchronization; for example, an attacker can block all messages to a receiver, but cannot prevent the receiver from advancing its clock from one epoch to the next.

For the evaluation of computational cost for signing, it is assumed that a sender caches all *non-intermediate* one-way chain elements–for BiBa and TV-HORS, this means *all* one-way chain elements ("keys" hereafter). *Intermediate* keys are only applicable to SCU+ and TSV+, and are defined in Section 5.3 and 5.4 respectively. In practice, a sender would employ algorithms like Coppersmith and Jakobsson's [6] to reduce the number of cached keys at the expense of higher signing cost, but our assumption is meant to put all schemes on an equal footing for comparison.

We emphasize that all formulas for $\mathscr{C}_\sigma$, $\mathscr{C}_v$, $\mathscr{L}_\sigma$ below have been validated with simulations.

In Algorithms 1 to 4 below, we denote a private key tuple by $(s_1, \ldots, s_t)$, a public key tuple by $(v_1, \ldots, v_t)$, a message by M, a counter by c, and a state tuple by $(S_1, \ldots, S_t)$.

5.1 BiBa

Algorithm 1 shows the BiBa MTS scheme. Our strategy is to determine $\mathscr{S}$, $\mathscr{C}_\sigma$, $\mathscr{C}_v$ and $\mathscr{L}_\sigma$ in turn.

Algorithm 1: The BiBa MTS scheme

$k \triangleq$ number of elements of a signature tuple
Key generation$(s_1, s_2, \ldots, s_t)$:
  $(v_1, v_2, \ldots, v_t) \leftarrow (\mathsf{PRF}(s_1, 0), \mathsf{PRF}(s_2, 0), \ldots, \mathsf{PRF}(s_t, 0))$
Signing$(M, s_1, s_2, \ldots, s_t)$:
  $c \leftarrow 0$
  repeat
    if $\exists I \subseteq \{1, \ldots, t\}$ s.t. $|I| = k$, $\mathsf{PRF}(H(M \| c), s_i)$ is the same $\forall i \in I$ **then**
      $\{i_1, i_2, \ldots, i_k\} \leftarrow I$
      return $(c, s_{i_1}, s_{i_2}, \ldots, s_{i_k})$
    end if
    $c \leftarrow c + 1$
  end repeat
Verification$(M, c, \sigma_1, \sigma_2, \ldots, \sigma_k)$:
  if $\sigma_i \neq \sigma_j$, $\forall i \neq j$, $1 \leq i, j \leq k$ **and**
    $\exists i \in \{1, \ldots, t\}$ s.t. $\mathsf{PRF}(\sigma_j, 0) = v_i$, $\forall j \in \{1, \ldots, k\}$ **and**
    $\mathsf{PRF}(H(M \| c), \sigma_j)$ is the same $\forall j \in \{1, \ldots, k\}$ **then**
    return "accept"
  else
    return "reject"
  end if

$\mathscr{S}$ and $\mathscr{C}_\sigma$ are related to P_S, an essential parameter of BiBa denoting "the probability that the sender can find a signature in one trial" [20], but this definition is imprecise. There are 4 ways by which the sender can find a signature:

1. increment c until there is exactly one bin with exactly k balls;
2. increment c until there is exactly one bin with at least k balls;
3. increment c until there is at least one bin with exactly k balls;
4. increment c until there is at least one bin with at least k balls.

Our validation of [20, Figure 5] suggests BiBa uses the 4th method above, but [20, Appendix A] indicates the 3rd method is used instead. To simplify our evaluation, we assume the 3rd method is used, i.e., $P_S \triangleq$ the probability of finding at least one bin with exactly k balls. P_S is related to $\mathscr{B}$ (defined in Lemma 1) as

$$P_S = \mathscr{B}(n, t, k)/n^t, \tag{1}$$

where n is the cardinality of the range of PRF. Consistent with our definition of P_S, the original security analysis of BiBa remains valid [20, p. 31], i.e.,

$$\mathscr{S} = \log_2 \frac{n^{rk-1}}{\binom{rk}{k}(n-1)^{rk-k}}. \tag{2}$$

LEMMA 1. *The number of ways to distribute t balls in n bins with at least one bin having exactly k balls is*

$$\mathscr{B}(n, t, k) \triangleq \sum_{i=1}^{\lfloor t/k \rfloor} \left[(-1)^{i-1} \binom{n}{i} (n-i)^{t-ik} \prod_{j=0}^{i-1} \binom{t-jk}{k} \right].$$

PROOF. Let A_i be the event that bin i has exactly k balls. $\bigcup_{i=1}^{n} A_i$ is the event that at least one bin has exactly k balls. Using the inclusion-exclusion principle, we have

$$|\bigcup_{i=1}^{n} A_i| = \sum_{i=1}^{n} |A_i| - \sum_{i,j} |A_i \cap A_j| + \cdots =$$
$$\binom{n}{1}\binom{t}{k}(n-1)^{t-k} - \binom{n}{2}\binom{t}{k}\binom{t-k}{k}(n-2)^{t-2k} + \cdots$$

There are at most $\lfloor t/k \rfloor$ bins with exactly k balls, so there are only $\lfloor t/k \rfloor$ terms in the expression above. □

To compute $\mathscr{C}_\sigma$ and $\mathscr{C}_v$, we replace all invocations of $\mathsf{PRF}(K, M)$ with $H(K\|M)$, which is valid provided H can be modelled as a random oracle [2]. Using (1), and [20, Table 2],

$$\mathscr{C}_\sigma = (1+t)n^t/\mathscr{B}(n,t,k). \tag{3}$$

During signature verification, a message is (i) hashed once together with a counter; (ii) the resultant hash is hashed with each signature element; and (iii) each signature element is verified against the last received signature element on the same one-way chain. The number of hash operations for item (iii) is given by Lemma 2. Therefore,

$$\mathscr{C}_v = 1 + k + \frac{k}{r[1-(1-1/t)^k]}. \tag{4}$$

LEMMA 2. *Denote by r the number of signatures generated per epoch. In uniform chain traversal, to verify a single signature element (of k in total), the expected number of hash operations required is $\left[r\left(1-(1-1/t)^k\right)\right]^{-1}$.*

PROOF. Without loss of generality, suppose a signature contains $s_{1,j}$. The corresponding public key element is one of $s_{1,0}, s_{1,1}, \ldots, s_{1,j-1}$, with "distances" from $s_{1,j}$ being $j, j-1, \ldots, 1$ respectively. Let A_d be the event that the distance is d ($d = 1, \ldots, j$). First, consider A_1, which only occurs if

1. $s_{1,j}$ has *not* been used in any signature in the current epoch yet: this has a probability of $\left(\frac{t-1}{t}\right)^{xk}$, where x is the number of signatures that have already been generated in the current epoch; and
2. $s_{1,j-1}$ has been used in a signature last epoch: this has a probability of $\left[1-\left(\frac{t-1}{t}\right)^{rk}\right]$.

So $\Pr[A_1] = \left(\frac{t-1}{t}\right)^{xk}\left[1-\left(\frac{t-1}{t}\right)^{rk}\right]$. Similarly, we have $\Pr[A_d] = \left(\frac{t-1}{t}\right)^{[x+(d-1)r]k}\left[1-\left(\frac{t-1}{t}\right)^{rk}\right]$. Now, the expected distance (conditioned on x) can be computed as

$$E[d|x] = \lim_{j\to\infty}\sum_{d=1}^{j} d\Pr[A_d]$$
$$= \left(\frac{t-1}{t}\right)^{(x-r)k}\left[1-\left(\frac{t-1}{t}\right)^{rk}\right]\lim_{j\to\infty}\sum_{d=1}^{j} d\left(\frac{t-1}{t}\right)^{drk}.$$

Substituting $q \stackrel{\triangle}{=} \frac{t-1}{t}$ and summing the infinite series in the expression above, we have

$$E[d|x] = q^{(x-r)k}(1-q^{rk})\frac{q^{rk}}{(1-q^{rk})^2} = \frac{q^{xk}}{1-q^{rk}},$$

and finally

$$E[d] = \sum_{x'=0}^{r-1} E[d|x=x']\Pr[x=x'] = \frac{1}{r(1-q^k)}.$$

□

A BiBa signature consists of a counter and k signature elements. Let the maximum value of a counter be c_{max}, then c_{max} happens at a probability of $\epsilon \stackrel{\triangle}{=} (1-P_S)^{c_{max}-1}P_S$. Note that $c_{max} \geq 1 \iff \epsilon \leq P_S$. If we want to use a short string to represent c_{max}, then we should keep ϵ low, e.g., 10^{-4}, and make sure $P_S \geq \epsilon$. With this consideration,

$$\mathscr{L}_\sigma = \lceil \log_2(c_{max}+1)\rceil + k\, l_H$$
$$= \left\lceil \log_2\left(\log_{1-P_S}\frac{\epsilon}{P_S}+2\right)\right\rceil + k\, l_H\,. \tag{5}$$

5.2 TV-HORS

Algorithm 2 shows the HORS/TV-HORS MTS scheme. Our strategy is to determine $\mathscr{C}_\sigma$, $\mathscr{L}_\sigma$, $\mathscr{C}_v$ and $\mathscr{S}$ in turn.

Algorithm 2: The HORS/TV-HORS MTS scheme

$k \stackrel{\triangle}{=}$ intended number of elements of a signature tuple
Key generation$(s_1, s_2, \ldots, s_t)$:
  $(v_1, v_2, \ldots, v_t) \leftarrow (H(s_1), H(s_2), \ldots, H(s_t))$
Signing$(M, s_1, s_2, \ldots, s_t)$:
  $(i_1, i_2, \ldots, i_k) \leftarrow \mathrm{Split}_k(H(M))$
  $\Sigma \leftarrow (s_{i_1}, s_{i_2}, \ldots, s_{i_k})$ with redundant elements removed
  return Σ
Verification(M, Σ):
  $(i_1, i_2, \ldots, i_k) \leftarrow \mathrm{Split}_k(H(M))$
  if $\exists\sigma \in \Sigma$ s.t. $H(\sigma) = v_i, \forall i \in \{i_1, i_2, \ldots, i_k\}$ **then**
    return "accept"
  else
    return "reject"
  end if

TV-HORS' signing cost is the same as HORS', i.e.,

$$\mathscr{C}_\sigma = 1.$$

Unlike BiBa, a HORS/TV-HORS signature may not always contain k distinct signature elements, because the signing function may produce redundant elements. According to Lemma 3,

$$\mathscr{L}_\sigma = \frac{l_H}{t^k}\sum_{i=1}^{k} i\,i!\binom{t}{i}\left\{{k \atop i}\right\}. \tag{6}$$

LEMMA 3. *The expected number of occupied bins if k balls are randomly thrown into t empty bins is $\frac{1}{t^k}\sum_{i=1}^{k} i\,i!\binom{t}{i}\left\{{k \atop i}\right\}$, where $\left\{{\cdot \atop \cdot}\right\}$ denotes a Stirling number of the second kind.*

PROOF. Let A_i be the event that i bins are occupied. There are $\binom{t}{i}$ ways to choose i from t empty bins, and $\left\{{k \atop i}\right\}$ ways to divide k balls into i bins. Furthermore, there are $i!$ ways to arrange the i chosen bins. Therefore, $\Pr[A_i] = \binom{t}{i}\left\{{k \atop i}\right\}i!/t^k$. $\sum_{i=1}^{k} i\Pr[A_i]$ gives us the expectation we need. □

Nevertheless, as $t/k \to \infty$, $\mathscr{L}_\sigma \to k$. Therefore to compute $\mathscr{C}_v$, we can re-use Lemma 2, i.e.,

$$\mathscr{C}_v = 1 + \frac{k}{r[1-(1-1/t)^k]}. \tag{7}$$

For estimating $\mathscr{S}$, suppose M_{att} is the message whose signature is to be forged. Let A_i denote the event that the attacker has captured i signature elements from r signatures; and B_j denote the event that $H(M_{att})$ requires j signature elements. The expected probability of forgery is

$$\sum_{j=1}^{k}\sum_{i=1}^{rk}\left(\frac{i}{t}\right)^j \Pr[A_i]\Pr[B_j],$$

where $\Pr[A_i]$ and $\Pr[B_j]$ are given by Lemma 3. Therefore,

$$\mathscr{S} = (rk + k)\log_2 t - \log_2 \sum_{i=1}^{rk}\sum_{j=1}^{k} \frac{i^j i!j!\binom{t}{i}\binom{t}{j}\left\{{rk \atop i}\right\}\left\{{k \atop j}\right\}}{t^j}. \quad (8)$$

When $t = 1024$, (8) requires $k \geq 14$ for at least 80 bits of security; whereas the widely used approximation [23] $\mathscr{S} = k\log_2 t - k\log_2(rk)$ requires $k \geq 13$.

5.3 SCU+

Algorithm 3 shows the SCU/SCU+ MTS scheme. Due to SCU's design, nonuniform chain traversal seems like a natural fit for SCU+, but as discussed in Section 5.1, uniform chain traversal is more robust and is used in SCU+ instead. Fig. 2 shows the epoch-j private key as $(s_{1,rj}, \ldots, s_{t,rj})$. All keys between $s_{i,r(j-1)}$ and $s_{i,rj}$, where $i = 1, \ldots, t$ and $j \geq 1$, are called *intermediate* keys. Our strategy is to determine $\mathscr{L}_\sigma$, $\mathscr{C}_\sigma$, $\mathscr{C}_v$ and $\mathscr{S}$ in turn.

Algorithm 3: The SCU/SCU+ MTS scheme

$(S_1, S_2, \ldots, S_t) \triangleq$ state tuple
Key generation$(s_1, s_2, \ldots, s_t)$:
  $S_i \leftarrow r$, $v_i \leftarrow H^r(s_i)$, $\forall i \in \{1, \ldots, t\}$
Signing$(M, s_1, s_2, \ldots, s_t)$:
  $c_z \leftarrow$ number of 0's in $H(M)$
  $I \leftarrow$ set of bit positions in $H(M)\|c_z$ where bit value is 1
  $S_i \leftarrow S_i - 1$, $\forall i \in I$
  return $(H^{S_i}(s_i) : i \in I)$
Verification$(M, \sigma_1, \sigma_2, \ldots, \sigma_k)$:
  $c_z \leftarrow$ number of 0's in $H(M)$
  $I \leftarrow$ set of bit positions in $H(M)\|c_z$ where bit value is 1
  if $k = |I|$ **and**
    $\exists i \in I$, $x_{ij} \in \mathbb{N}^+$, s.t. $H^{x_{ij}}(\sigma_j) = v_i$, $\forall j \in \{1, \ldots, k\}$ **then**
    $v_i \leftarrow \sigma_j$, $\forall H^{x_{ij}}(\sigma_j) = v_i$
    return "accept"
  else
    return "reject"
  end if

$\mathscr{L}_\sigma$ is proportional to the expected number of 1-bits in $H(M)\|c_z$. In $H(M)$, the expected number of 1-bits is $l_\mathcal{H}/2$ (we distinguish $l_\mathcal{H}$ from the normal hash length l_H because typically $l_\mathcal{H} \geq l_H$ by design). In c_z, the expected number of 1-bits varies with $l_\mathcal{H}$, because c_z may be longer than necessary to represent the number of 0's in $H(M)$. In fact, c_z is of length

$$l_c \triangleq |c_z| = \lceil \log_2 (l_\mathcal{H} + 1) \rceil, \quad (9)$$

and by t's definition, $t = l_\mathcal{H} + l_c$. Fig. 3 shows the probability of having 1 at the ith bit of c_z for $l_\mathcal{H} = 128, \ldots, 248$. For the case $l_\mathcal{H} = 160$ (the length of a SHA-1 or RIPEMD-160 hash value), bits 4-8 are 1 at a probability of 1/2, bit 2 is almost always 1, bit 3 is almost always 0, and bit 1 is always 0; in other words,

$$\mathscr{L}_\sigma = \left[l_\mathcal{H}\frac{1}{2} + (l_c - 3)\frac{1}{2} + 1 \right] l_H = \frac{t-1}{2} l_H.$$

For the general case, it is simpler to use the approximation

$$\mathscr{L}_\sigma = t\, l_H/2. \quad (10)$$

Signing cost varies with the state variables corresponding to the 1-bits of $H(M)\|c_z$. Without loss of generality, let us consider the first bit of $H(M)\|c_z$. Within an

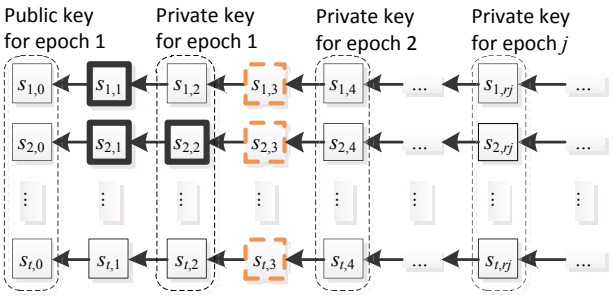

Figure 2: SCU+ with uniform chain traversal and $r = 2$. Suppose in epoch 1, two signatures are received: $(s_{1,1}, s_{2,1})$ and $(s_{2,2})$ (note "thick boxes"). In epoch 2, to verify signature $(s_{1,3}, s_{2,3}, s_{t,3})$ (note "orange dashed boxes"), a receiver checks $H^2(s_{1,3}) \stackrel{?}{=} s_{1,1}$, $H(s_{2,3}) \stackrel{?}{=} s_{2,2}$, and $H^3(s_{t,3}) \stackrel{?}{=} s_{t,0}$.

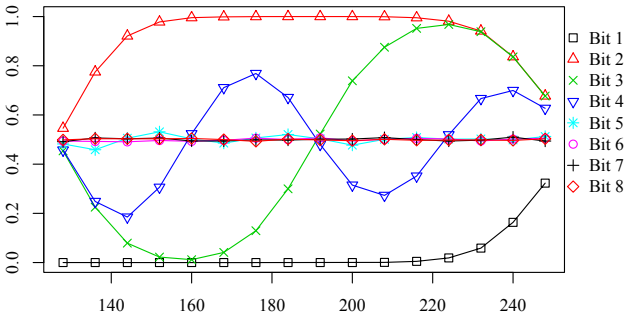

Figure 3: Y-axis: probability of having bit value 1 at the ith bit of c_z. X-asix: $l_\mathcal{H}$.

epoch, if the first bit is 1 once, which occurs at a probability of $\binom{r}{1}(\frac{1}{2})(\frac{1}{2})^{r-1}$, then the accumulative signing cost is $r-1$. If the first bit is 1 twice, which occurs at a probability of $\binom{r}{2}(\frac{1}{2})^2(\frac{1}{2})^{r-2}$, then the accumulative signing cost is $(r-1)+(r-2)$. By extension, the expected signing cost for the first bit is given by

$$\frac{1}{2^r}\sum_{i=1}^{r}\left[\binom{r}{i}\sum_{j=1}^{i}(r-j)\right] = \frac{3r(r-1)}{8}.$$

The expected signing cost for all bits is then

$$\mathscr{C}_\sigma = 3tr(r-1)/16. \quad (11)$$

When a signature is received, each of the $t/2$ (on average) signature elements needs to be verified. Without loss of generality, let us consider a signature element corresponding to the first bit of $H(M)\|c_z$. Let A_d be the event that this signature element requires d hash operations to verify, which occurs when the past $d-1$ signatures do not contain a signature element corresponding to the first bit of $H(M)\|c_z$, *but* the dth signature in the past does, i.e., $\Pr[A_d] = \frac{1}{2}(1-\frac{1}{2})^{d-1}$, and

$$\mathscr{C}_v = 1 + \frac{t}{2}\sum_{d=1}^{\infty} d\Pr[A_d] = 1 + t. \quad (12)$$

To determine $\mathscr{S}$, we estimate the success probability of signature forgery during epoch j. To forge a signature for M_{att}, an attacker needs to supply $s_{i_1,j}, \ldots, s_{i_k,j}$, where i_1,

$\ldots, i_k$ correspond to the positions of 1-bits in $H(M_{att})\|c_z$. Suppose the attacker has already captured r signatures for epoch j. The success probability of signature forgery, Pr[forgery|k], is the probability that bit positions $i_1, \ldots, i_k$ are covered by a subset of the r captured signatures. For the case of $l_{\mathcal{H}} = 160$, there is almost always a bit position among $i_1, \ldots, i_k$ that corresponds to bit 2 in c_z (see Fig. 3), so the attacker only has to match $k-1$ bits to the bit positions that are already compromised, i.e., $\Pr[\text{forgery}] = (1 - 1/2^r)^{k-1}$. For the general case, it is simpler to use the approximation $\Pr[\text{forgery}] = (1 - 1/2^r)^k$. Next, let us consider the probability of having k 1-bits in $H(M_{att})\|c_z$, denoted by $\Pr[k]$. If we denote by A_i the event that $H(M_{att})$ has i 1-bits, and by B_i the event that c_z has i 1-bits, then $\Pr[k] = \sum_{i=0}^{\min(k,l_{\mathcal{H}})} \Pr[A_i] \Pr[B_{k-i}|A_i]$. Therefore,

$$\begin{aligned}
\Pr[\text{forgery}] &= \sum_{k'=1}^{t} \Pr[\text{forgery}|k = k'] \Pr[k = k'] \\
&= \sum_{k'=1}^{t} \left[(1 - 1/2^r)^{k'} \sum_{i=0}^{\min(k', l_{\mathcal{H}})} \Pr[A_i] \Pr[B_{k'-i}|A_i] \right] \\
&= \sum_{k'=1}^{t} \left[(1 - 1/2^r)^{k'} \sum_{i=0}^{\min(k', l_{\mathcal{H}})} \frac{\binom{l_{\mathcal{H}}}{i}}{2^{l_{\mathcal{H}}}} I(l_c, l_{\mathcal{H}} - i, k' - i) \right],
\end{aligned}$$

and

$$\begin{aligned}
\mathscr{S} = l_{\mathcal{H}} - \log_2 \Bigg\{ \sum_{k'=1}^{t} \Bigg[(1 - 1/2^r)^{k'} \\
\sum_{i=0}^{\min(k', l_{\mathcal{H}})} \binom{l_{\mathcal{H}}}{i} I(l_c, l_{\mathcal{H}} - i, k' - i) \Bigg] \Bigg\},
\end{aligned} \tag{13}$$

where $I(l_c, l_{\mathcal{H}} - i, k' - i)$ is defined by Definition 3.

DEFINITION 3. *$I(l_c, b_1, b_2)$ is 1 if the following has a solution, and 0 if otherwise:*

$$\begin{bmatrix} 2^{l_c-1} & 2^{l_c-2} & \cdots & 2^0 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_{l_c-1} \\ \vdots \\ x_0 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

$$x_0, \ldots, x_{l_c-1} \in \{0, 1\}.$$

Note the system of equations above does not always have a solution, e.g., when $l_c \geq b_1 = b_2 = 2$. A closed-form expression for $I(l_c, b_1, b_2)$ is unknown.

5.4 TSV+

TSV+ inherits TSV's most notable features:

- For the same k, TSV becomes more secure than HORS by imposing an order/sequence on the signature elements.
- For efficiency, the order is imposed on individual groups, and not across all signature elements.
- So that signature elements are not interchangeable between groups, TSV releases keys at different levels of the one-way chains depending on the group.

However, TSV+ introduces two main enhancements: firstly, so that it is comparable to other MA schemes, we enable TSV+ to support multiple signatures within an epoch; secondly, TSV+ uses uniform chain traversal because it is more robust than nonuniform chain traversal (which is used in TSV), as we explained in Section 4. As shown in Algorithm 4, TSV+ uses a state tuple (like SCU/SCU+ does) to keep track of intermediate keys (between a public/private key pair or a pair of adjacent private keys). The number of intermediate keys is $(wg - 1)$, as shown in Fig. 4, where w is by design the smallest integer such that the probability of a one-way chain being used for more than w out of r signatures in an epoch is less than 10^{-4}. The probability of a one-way chain being used is $\binom{t-1}{k-1}/\binom{t}{k} = \frac{k}{t}$, so w is the smallest integer such that

$$\sum_{i=w+1}^{r} \binom{r}{i} \left(\frac{k}{t}\right)^i \left(1 - \frac{k}{t}\right)^{r-i} < 10^{-4}. \tag{14}$$

Fig. 4 shows the epoch-j private key as $(s_{1,wgj}, \ldots, s_{t,wgj})$, and in that example, $w = 2$. For analysis, our strategy is to first determine $\mathscr{L}_\sigma$ and $\mathscr{C}_\sigma$, which are closely related; then $\mathscr{C}_v$ and $\mathscr{S}$.

Algorithm 4: The TSV+ MTS scheme

$k \triangleq$ number of elements of a signature tuple
$g \triangleq$ number of groups
$n_i \triangleq$ number of $\log_2 t$-bit strings in group i, $\forall i = 1, \ldots, g$, under the constraint $\sum_{i=1}^{g} n_i = k$
$w \triangleq$ smallest integer that satisfies (14)
$(S_1, \ldots, S_t) \triangleq$ state
Key generation$(s_1, s_2, \ldots, s_t)$:
  $S_i \leftarrow wg$, $v_i \leftarrow H^{wg}(s_i)$, $\forall i \in \{1, \ldots, t\}$
Signing$(M, s_1, s_2, \ldots, s_t)$:
  $c \leftarrow 0$
  repeat
    $(h_1, h_2, \ldots, h_g) \leftarrow \text{Split}_g(H(M\|c))$
    $(i_1, \ldots, i_{n_1}) \leftarrow \text{Split}_{n_1}(h_1)$
    $(i_{n_1+1}, \ldots, i_{n_1+n_2}) \leftarrow \text{Split}_{n_2}(h_2)$
    $\cdots$
    $(i_{k-n_g+1}, \ldots, i_k) \leftarrow \text{Split}_{n_g}(h_g)$
    if each of $h_1, \ldots, h_g$ consists of decreasing elements **and** $i_1, \ldots, i_k$ are distinct **then**
      $S_{i_j} \leftarrow S_{i_j} - 1$, $\forall j \in \{1, \ldots, n_1\}$
      $S_{i_j} \leftarrow S_{i_j} - 2$, $\forall j \in \{n_1 + 1, \ldots, n_1 + n_2\}$
      $\cdots$
      $S_{i_j} \leftarrow S_{i_j} - g$, $\forall j \in \{k - n_g + 1, \ldots, k\}$
      return $(c, H^{S_{i_1}}(s_{i_1}), \ldots, H^{S_{i_k}}(s_{i_k}))$
    end if
    $c \leftarrow c + 1$
  end repeat
Verification$(M, c, \sigma_1, \sigma_2, \ldots, \sigma_k)$:
  $(h_1, h_2, \ldots, h_g) \leftarrow \text{Split}_g(H(M\|c))$
  $(i_1, \ldots, i_{n_1}) \leftarrow \text{Split}_{n_1}(h_1)$
  $(i_{n_1+1}, \ldots, i_{n_1+n_2}) \leftarrow \text{Split}_{n_2}(h_2)$
  $\cdots$
  $(i_{k-n_g+1}, \ldots, i_k) \leftarrow \text{Split}_{n_g}(h_g)$
  if each of $h_1, \ldots, h_g$ consists of decreasing elements **and** $i_1, \ldots, i_k$ are distinct **and** $\exists i \in \{i_1, \ldots, i_k\}$, $x_{ij} \in \mathbb{N}^+$, s.t. $H^{x_{ij}}(\sigma_j) = v_i$, $\forall j \in \{1, \ldots, k\}$ **then**
    $v_i \leftarrow \sigma_j$, $\forall\, H^{x_{ij}}(\sigma_j) = v_i$
    return "accept"
  else
    return "reject"
  end if

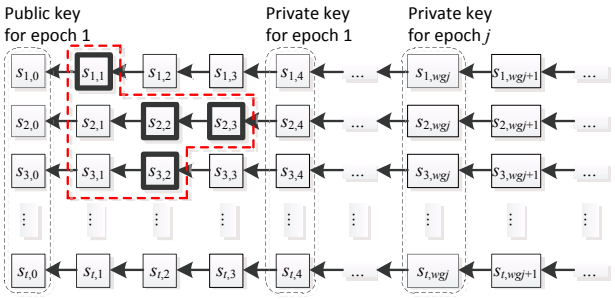

Figure 4: TSV+ with uniform chain traversal, and $k = g = w = 2$, $n_1 = n_2 = 1$, so $wg = 4$ (see Algorithm 4 for definition of symbols). In this example, suppose corresponding to the first message M_1, $H(M_1 \| c_1) = 1\|2$, so the first signature is $(H^{4-1}(s_{1,4}), H^{4-2}(s_{2,4})) = (s_{1,1}, s_{2,2})$. Suppose corresponding to the second message M_2, $H(M_2 \| c_2) = 2\|3$, so the second signature is $(H^{2-1}(s_{2,4}), H^{4-2}(s_{3,4})) = (s_{2,3}, s_{3,2})$.

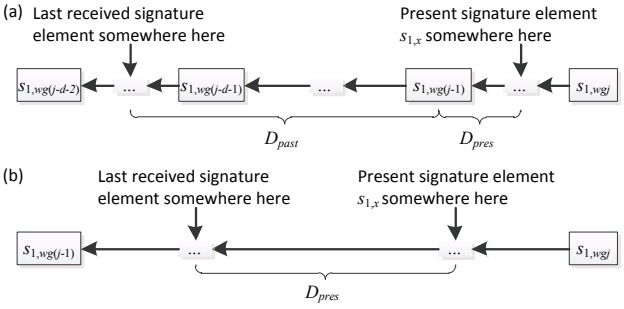

Figure 5: Two cases to be considered for the derivation of $\mathscr{C}_v$ for TSV+.

The probability of finding an acceptable c is correctly given by $P_S \triangleq k!\binom{t}{k}/(t^k \prod_{i=1}^{g} n_i!)$ [15]. Using the same reasoning for (5),

$$\mathscr{L}_\sigma = \left\lceil \log_2 \left(\log_{1-P_S} \frac{\epsilon}{P_S} + 2 \right) \right\rceil + k\, l_H,$$

where ϵ is a small user-defined constant, e.g., 10^{-4}. After finding c, the signer invokes H at a multiplicity that depends on the state tuple $(S_1, \ldots, S_t)$ at the last line of Algorithm 4 – we need to calculate the expected value of this multiplicity. Let $A_{g_1,g_2,\ldots,g_i}$ denote the event that a one-way chain is used for i out of r signatures in an epoch, and for the i signatures, the one-way chain belongs to groups $g_1, g_2, \ldots, g_i$ respectively. Event $A_{g_1,g_2,\ldots,g_i}$ occurs at a probability of

$$\binom{r}{i} \left(\frac{k}{t}\right)^i \left(1 - \frac{k}{t}\right)^{r-i} \frac{n_{g_1} n_{g_2} \cdots n_{g_i}}{k^i},$$

during which the *expected* number of hash invocations is

$$\frac{(wg - g_1) + (wg - g_1 - g_2) + \cdots + (wg - g_1 - g_2 - \cdots - g_i)}{r}.$$

Since there are t one-way chains,

$$\mathscr{C}_\sigma = \frac{1}{P_S} + t \sum_{i=1}^{r} \binom{r}{i} \left(\frac{k}{t}\right)^i \left(1 - \frac{k}{t}\right)^{r-i} \sum_{g_1=1}^{g} \cdots \sum_{g_i=1}^{g} \frac{n_{g_1} \cdots n_{g_i}}{k^i} \frac{(wg - g_1) + \cdots + (wg - g_1 - g_2 - \cdots - g_i)}{r}. \quad (15)$$

For the case $n_1 = n_2 = \cdots = n_g = k/g$, (15) can be simplified as

$$\mathscr{C}_\sigma = \frac{1}{P_S} + \frac{t}{r} \sum_{i=1}^{r} i \binom{r}{i} \left(\frac{k}{t}\right)^i \left(1 - \frac{k}{t}\right)^{r-i} \cdots \left[\left(w - \frac{i+1}{4}\right) g - \frac{i+1}{4}\right] = \frac{1}{P_S} + kwg + k(g+1)\left(\frac{k - kr}{4t} - \frac{1}{2}\right). \quad (16)$$

(15) can be further simplified when k/t and r are small. In this case, the probability that a one-way chain is used more than once in an epoch is negligible. In an epoch, a one-way chain is chosen at a probability of k/t, and when chosen, it belongs to one of g groups. The probability of the one-way chain belonging to group i is given by n_i/k $(i = 1, \ldots, g)$. Therefore,

$$\mathscr{C}_\sigma = \frac{1}{P_S} + t \left[\left(1 - \frac{k}{t}\right) \cdot 0 + \sum_{i=1}^{g} \frac{k}{t} \frac{n_i}{k} (wg - i) \right] = \frac{1}{P_S} + \sum_{i=1}^{g} n_i (wg - i). \quad (17)$$

Deriving $\mathscr{C}_v$ is more involved. Suppose the present signature element $s_{1,x}$, which without loss of generality, falls on the first one-way chain. If $s_{1,x}$ belongs to the *first* signature in an arbitrary epoch j, then by definition, the last received signature element on the same one-way chain must be from a past epoch. The expected "distance" between $s_{1,x}$ and the last received signature element (see Fig. 5(a)) is the sum of

- $D_{past} \triangleq$ the expected distance between the last received signature element and $s_{1,wg(j-1)}$; and
- $D_{pres} \triangleq$ the expected distance between $s_{1,wg(j-1)}$ and $s_{1,x}$.

So when $s_{1,x}$ belongs to a first signature, which occurs at a probability of $1/r$, the distance is $D_{past} + D_{pres}$.

Now, suppose $s_{1,x}$ belongs to the *second* signature in epoch j. The last received signature element on the same one-way chain can either be

1. from a past epoch, at a probability of $q \triangleq 1 - k/t$; or
2. from the current epoch, at a probability of $1 - q$.

For the first case, we showed that the expected distance is $D_{past} + D_{pres}$, whereas for the second case, the expected distance is D_{pres} (see Fig. 5(b)). So when $s_{1,x}$ belongs to a second signature, which occurs at a probability of $1/r$, the expected distance is $(D_{past} + D_{pres})q + D_{pres}(1 - q)$.

Applying the reasoning above to the cases that $s_{1,x}$ belongs to the third signature, the fourth signature and so on,

we can write the expected distance between $s_{1,x}$ and the last received signature element as

$$\frac{1}{r}\sum_{i=1}^{r}\{(D_{past}+D_{pres})q^{i-1}+D_{pres}(1-q^{i-1})\}$$
$$=\frac{(1-q^r)}{r(1-q)}D_{past}+D_{pres}. \quad (18)$$

The estimation of $\mathscr{C}_v$ is now reduced to the estimation of D_{past} and D_{pres} in (18). D_{pres} is simply

$$D_{pres}=\frac{1}{g}\sum_{i=1}^{g} i=\frac{g+1}{2}. \quad (19)$$

To find D_{past}, let $A_{d,g_1,g_2,\ldots,g_i}$ denote the event that

- the past epoch and the present epoch in Fig. 5(a) are separated by d epochs of no signature, where $d \geq 0$;
- and in the past epoch, i signature elements have been received, which belong to group $g_1, g_2, \ldots, g_i$ respectively.

Event $A_{d,g_1,g_2,\ldots,g_i}$ occurs at a probability of

$$\binom{r}{i}\left(\frac{k}{t}\right)^i\left(1-\frac{k}{t}\right)^{(d+1)r-i}\frac{n_{g_1}n_{g_2}\cdots n_{g_i}}{k^i},$$

during which the distance is

$$dwg+(wg-g_1-g_2-\cdots-g_i).$$

Therefore,

$$D_{past}=\sum_{d=0}^{\infty}\sum_{i=1}^{r}\sum_{g_1=1}^{g}\cdots\sum_{g_i=1}^{g} [(d+1)wg-g_1-\cdots-g_i]\Pr[A_{d,g_1,g_2,\ldots,g_i}]. \quad (20)$$

When $n_1=n_2=\cdots=n_g=k/g$, (20) can be simplified as

$$D_{past}=\sum_{d=0}^{\infty}\sum_{i=1}^{r}\binom{r}{i}\left(\frac{k}{t}\right)^i\left(1-\frac{k}{t}\right)^{(d+1)r-i}\left[(d+1)wg-\frac{i}{2}g-\frac{i}{2}\right]$$
$$=\frac{2twg-kr(g+1)}{2t[1-(1-k/t)^r]}. \quad (21)$$

Substituting (19) and (21) back into (18), the expected distance between $s_{1,x}$ and the last received signature element becomes simply $\frac{twg}{kr}$. Since a signature has k signature elements,

$$\mathscr{C}_v=twg/r, \quad (22)$$

for the special case of uniform group sizes.

Now, we look at $\mathscr{S}$. In Fig. 4 where $k=2$, we can see that if an attacker (i) manages to capture the signature elements marked by thick frames, namely $s_{1,1}$, $s_{2,2}$, $s_{2,3}$ and $s_{3,2}$, and (ii) block these signature elements from the recipients, then the attacker can forge a signature using *any* two of the elements surrounded by the red dashed contour. In reality, r captured signatures use at most rk distinct one-way chains, but for small k/t and r, it is approximately true that r captured signatures use exactly rk distinct one-way chains (and we used the same approximation for (17)). Let us denote by $\mathcal{G}_i$ the set of one-way chains corresponding to the captured signature elements of groups $i, i+1, \ldots, g$ ($i=1,\ldots,g$). Hence, $\mathcal{G}_g$ has rn_g elements, $\mathcal{G}_{g-1}$ has $r(n_{g-1}+n_g)$ elements, and so on. An attacker successfully forges a signature if he/she is able to ensure the forged group-g signature elements lie on any n_g one-way chains from the set $\mathcal{G}_g$; the forged group-$(g-1)$ signature elements lie on n_{g-1} one-way chains from the set $\mathcal{G}_{g-1}$ that are distinct from previously chosen one-way chains; and so on. In other words, the number of ways to forge a signature is

$$\binom{rn_g}{n_g}\binom{rn_{r-1}+(r-1)n_g}{n_{g-1}}\cdots\binom{rn_1+(r-1)(n_g+n_{g-1}+\cdots+n_2)}{n_1}$$
$$=\binom{rn_g}{n_g}\prod_{i=1}^{g-1}\binom{rn_i+(r-1)\sum_{j=i+1}^{g}n_j}{n_i}. \quad (23)$$

When $r=1$, (23) reduces to 1, consistent with intuition. Therefore,

$$\mathscr{S}=\log_2\frac{\binom{t}{k}}{\binom{rn_g}{n_g}\prod_{i=1}^{g-1}\binom{rn_i+(r-1)\sum_{j=i+1}^{g}n_j}{n_i}}. \quad (24)$$

When $r=1$, (24) becomes $\log_2\binom{t}{k}$, which is different from Li and Cao's $k\log_2 t$, because they consider $H(M_{att})$ instead of $H(M_{att}\|c_{att})$.

6. APPLICATION TO WIDE-AREA MEASUREMENT SYSTEMS

The analysis in the previous section forms the basis for comparison of BiBa, TV-HORS, SCU+ and TSV+. For the comparison to be performed in the context of the wide-area measurement system (WAMS), an introduction to the WAMS is given here. A WAMS is essentially a high-speed network of **phasor measurement units (PMUs)**, whose sole objective is to report voltage and current phasor measurements (amplitude, frequency and phase). Given enough real-time phasors, the state of the grid (voltage and phase angle of each bus) can be tracked, giving the utility enhanced "situational awareness" about its system. This enhanced situational awareness provides many advantages: improved operation planning, optimized transmission assets utilization, system stabilization, disturbances containment, etc. In fact, the lack of this level of situational awareness is one of the factors that contributed to the infamous 2003 North America and 2003 Italy blackouts [28, 30].

The WAMS consists of four components: (i) synchronized PMUs (also called *synchrophasors*), (ii) phasor data concentrators (PDCs), (iii) wide area network (WAN), and (iv) real-time database and data archiver [16]. Fig. 6 shows the four-layer generic architecture of the WAMS [16]. The PMUs in Layer 1 report voltage and current phasors that are time-stamped with high-precision internal clocks and the Global Positioning System at 10-30 frames per second, enabling the correlation of phasor measurements across a wide grid area. The PMUs transmit the data in the IEEE C37.118 format to the PDCs in Layer 2 via the WAN. The PDCs correlate the time-tagged data, and forward the data to the Applications Data Buffer in Layer 3. The Applications Data Buffer monitors the data for losses, errors and

synchronization, in addition to supplying the data in the required format to the applications in Layer 4. Layer 4 consists of the Real Time Database and Data Archiver, which is responsible for collecting and archiving data for post-incident analysis and assessment. Layer 4 also contains applications for monitoring, control and protection functions.

PMUs are required to multicast phasor data to multiple consumers including PDCs for communication redundancy, whereas PDCs at the same hierarchical level are required to share data with each other through multicast [1].

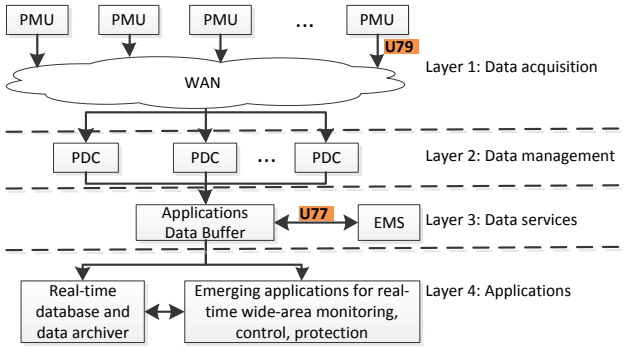

Figure 6: Generic architecture of the WAMS. U77 and U79 are logical interfaces defined in NISTIR 7628 [18].

7. COMPARING MULTICAST AUTHENTICATION SCHEMES

With application to the WAMS in mind, the MA schemes BiBa, TV-HORS, SCU+ and TSV+ are evaluated in terms of the metrics (i) $\mathscr{L}_\sigma \,/\, \mathscr{S}$, (ii) $\mathscr{C}_\sigma \,/\, \mathscr{S}$, and (ii) $\mathscr{C}_v \,/\, \mathscr{S}$; and compared with each other. The parameters of each scheme are set under the following constraints:

- **Security level**: Each scheme must provide a security level of *at least* 80 bits.
- **Signature length**: A recent simulation study [9] suggests that a signature should be *at most* 300 bytes, because a C37.118 frame can be as much as 1200 bytes long while a WAN typically supports a maximum transmission unit (MTU) of 1500 bytes.
- **Hash length**: Since second preimage resistance is weaker than preimage resistance, we are primarily concerned with the former. For any truncated hash of SHA-1, SHA-224, SHA-256 and SHA-512, the actual second preimage resistance is influenced by the preimage length, but an 80-bit truncated SHA-384 hash has a second preimage resistance of exactly 80 bits [7]. So, we set $l_H = 80$ assuming SHA-384 hashes are truncated to 80 bits.
- **Number of one-way chains**: With the exception of SCU+, we fix $t = 1024$ following standard practice [15, 20, 23, 31].

By default, we configure the parameters according to Table 1 to satisfy the constraints above, as well as to minimize the signature length due to the large data volume in a WAMS.

For BiBa, two configurations are provided: BiBa_0 is the default configuration, whereas BiBa_1 satisfies the additional constraint $\mathscr{C}_\sigma \leq 10\,\mathscr{C}_v$. Compared to BiBa_0, BiBa_1 trades off communication efficiency for signing efficiency, but cannot support $r \geq 4$. For SCU+ and TSV+, there are no suitable parameter values that satisfy the above constraints for $r \geq 3$. Two TSV+ configurations are provided: TSV+_0 is the default configuration, whereas TSV+_1 satisfies the additional constraints $\mathscr{C}_\sigma \leq 10\,\mathscr{C}_v$ and $\mathscr{C}_v \leq 10\,\mathscr{C}_\sigma$ (i.e., $\mathscr{C}_\sigma$ and $\mathscr{C}_v$ are at most one order of magnitude different from each other). Compared to TSV+_0, TSV+_1 is meant to provide more balanced signing and verification costs.

Table 1: Configurations used for comparison, found through exhaustive search.

Configuration	$r=1$	2	3	4	5
BiBa_0 (k,n)	9, 1414	12, 618	15, 358	19, 215	24, 137
BiBa_1 (k,n)	11, 256	17, 123	24, 72		
TV-HORS (k)	14	18	21	25	29
SCU+ $(l_{\mathcal{H}})$	185	405			
TSV+_0 (k,g,n,w)	11, 11, $\{1,1,1,1,1,1,1,1,1,1,1\}$, 1				
		18, 9, $\{4,4,3,2,1,1,1,1,1\}$, 1			
TSV+_1 (k,g,n,w)	11, 3, $\{2,3,6\}$, 1				
		18, 9, $\{4,4,3,2,1,1,1,1,1\}$, 1			

Fig. 7 to 9 show the bar charts for $\mathscr{L}_\sigma \,/\, \mathscr{S}$, $\mathscr{C}_\sigma \,/\, \mathscr{S}$, and $\mathscr{C}_v \,/\, \mathscr{S}$. The charts show only $r = 1, 2$ since not all configurations support $r \geq 3$. BiBa_0 is the best performer in signature length but has far poorer efficiency in signing than the others. BiBa_1 produces slightly longer signatures but has significantly better signing efficiency than BiBa_0, so BiBa_1 is more practical. SCU+ is efficient in signing and verification but requires far longer signatures than the others for the same security level. TSV+ (both TSV+_0 and TSV+_1) is more efficient than TV-HORS in signature length when $r = 1$, but not when $r = 2$; moreover, TSV+ is several orders of magnitude slower than TV-HORS in signing and verification. Compared to TSV+_0, TSV+_1 has more balanced signing and verification costs, and is hence better for senders and receivers with similar capabilities. Despite its algorithmic simplicity, TV-HORS is a good performer in all categories. Although proposed after BiBa, SCU and TSV do not offer clear advantages over BiBa.

8. CONCLUSION AND FUTURE WORK

This work is motivated by the need for an efficient multicast authentication (MA) scheme to secure the required real-time multicast traffic within a wide-area measurement system (WAMS). For real-time systems like the WAMS, an MA scheme is best constructed from a multiple-time signature (MTS) scheme rather than a conventional digital signature scheme [15, 31]. Instead of designing yet another MTS-based MA scheme from scratch, this work executes the common sense of first attempting to find suitable candidates among the many MTS-based MA schemes already proposed to date. To this end, we first identify four representative MA schemes, namely BiBa, TV-HORS, SCU+ and TSV+. Among these MA schemes, SCU+ is an MA scheme we constructed from an MTS scheme designed for secure code update [29], and TSV+ is our patched version of TSV [15], an MA scheme which we show to be vulner-

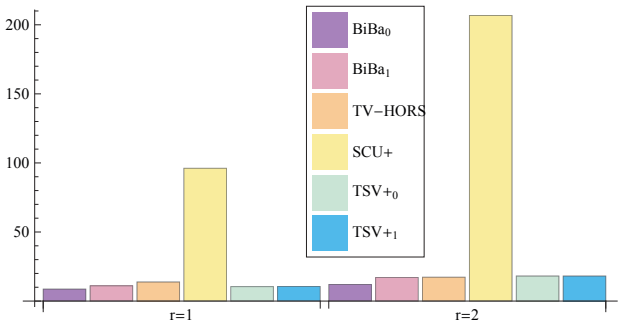

Figure 7: A plot of $\mathscr{L}_\sigma$ / $\mathscr{S}$ against r. Lower is better.

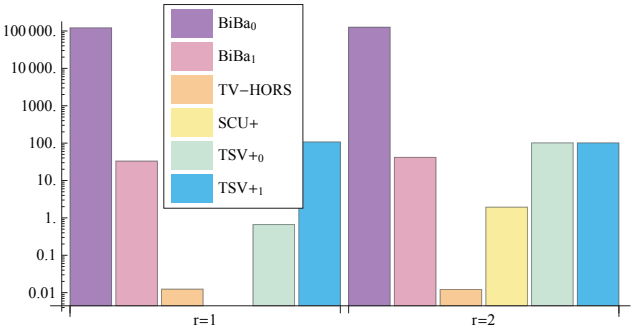

Figure 8: A plot of $\mathscr{C}_\sigma$ / $\mathscr{S}$ against r. Lower is better.

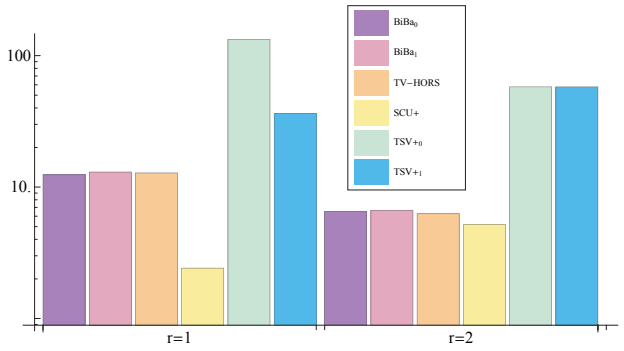

Figure 9: A plot of $\mathscr{C}_v$ / $\mathscr{S}$ against r. Lower is better.

able. We then provide rigorous mathematical analysis of these schemes. Our simulation-validated analysis fills the gaps of, and at places rectifies or improves existing analyses. Based on our analysis, our comparison shows that TV-HORS, while algorithmically unsophisticated, has the most balanced computational and communication efficiencies relative to security levels. SCU+, TSV+ and by extension SCU and TSV do not offer clear advantages over BiBa, the oldest among the studied schemes. As a follow-up to this preliminary study, we aim to expand our analysis and comparison to cover more schemes. Theoretical accounting of memory costs is nontrivial and will be attempted in future work. Just as naming a single superior MTS scheme is nontrivial [25], naming a single superior MA scheme is equally nontrivial. This preliminary work serves as a first step, and already we know that TV-HORS has set a benchmark.

Acknowledgment

The authors would like to thank Dr Gina Kounga and Dr Anthony Lo for reviewing an early draft of this paper, and Prof Ahmad-Reza Sadeghi for shepherding this paper. Yee Wei Law is partly supported by the Institute for a Broadband-Enabled Society, the ARC under the Discovery Project grant DP1095452, and the EC under contract FP7-ICT-2009-257992 "SmartSantander". GONG Zheng is supported by NSFC (61100201, 61070217), Foundation for Distinguished Young Talents in Higher Education of Guangdong (LYM11053), and Guangzhou Science and Technology Plan Project (11C42090777).

9. REFERENCES

[1] M. Adamiak, B. Kasztenny, and W. Premerlani. Synchrophasors: definition, measurement, and application. In *59th Annual Georgia Tech Protective Relaying*, 2005.

[2] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[3] R. Bobba, H. Khurana, M. AlTurki, and F. Ashraf. PBES: a policy based encryption system with application to data sharing in the power grid. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 262–275, New York, NY, USA, 2009. ACM.

[4] P. G. Bradford and O. V. Gavrylyako. Foundations of security for hash chains in ad hoc networks. *Cluster Computing*, 8(2):189–195, 2005.

[5] P. G. Bradford and O. V. Gavrylyako. Hash chains with diminishing ranges for sensors. *Int. J. High Performance Computing and Networking*, 4(1/2):31–38, 2006.

[6] D. Coppersmith and M. Jakobsson. Almost optimal hash sequence traversal. In *Financial Cryptography*, volume 2357 of *Lecture Notes in Computer Science*, pages 102–119. Springer Berlin / Heidelberg, 2003.

[7] Q. Dang. Recommendation for applications using approved hash algorithms. NIST Special Publication 800-107, Computer Security Division, Information Technology Laboratory, NIST, Feb. 2009.

[8] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. Inf. Theory*, 29(2):198–208, Mar. 1983.

[9] P. Kansal and A. Bose. Smart grid communication requirements for the high voltage power system. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–6, July 2011.

[10] J. Katz. *Digital Signatures*. Springer, 2010.

[11] I. Krontiris and T. Dimitriou. Authenticated in-network programming for wireless sensor networks. *Ad-Hoc, Mobile, and Wireless Network*, 4104:390–403, 2006.

[12] L. Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI Intl. Computer Science Laboratory, Oct. 1979.

[13] Y. W. Law, M. Palaniswami, G. Kounga, and A. Lo. WAKE: Key Management Scheme for Wide-Area

Measurement Systems in Smart Grid. *IEEE Communications Magazine*, Jan. 2013, in press.

[14] J. Lee, S. Kim, Y. Cho, Y. Chung, and Y. Park. HORSIC: An efficient one-time signature scheme for wireless sensor networks. *Information Processing Letters*, 112(20):783–787, 2012.

[15] Q. Li and G. Cao. Multicast authentication in the smart grid with one-time signature. *IEEE Transactions on Smart Grid*, 2(4):686–696, 2011.

[16] C. Martinez, M. Parashar, J. Dyer, and J. Coroas. Phasor Data Requirements for Real Time Wide-Area Monitoring, Control and Protection Applications. White paper, EIPP – Real Time Task Team, Jan. 2005.

[17] C. Meadows and P. Syverson. Formalizing GDOI group key management requirements in NPATRL. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 235–244. ACM, 2001.

[18] NIST. Guidelines for smart grid cyber security. IR 7628, Aug. 2010.

[19] Y. Park and Y. Cho. Efficient one-time signature schemes for stream authentication. *Journal of Information Science and Engineering*, 22(3):611–624, 2006.

[20] A. Perrig. The BiBa one-time signature and broadcast authentication protocol. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 28–37. ACM, 2001.

[21] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proceedings of the 7th Ann. Int. Conf. on Mobile Computing and Networking*, pages 189–199. ACM Press, 2001.

[22] J. Pieprzyk, H. Wang, and C. Xing. Multiple-time signature schemes against adaptive chosen message attacks. In *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 88–100. Springer Berlin / Heidelberg, 2004.

[23] L. Reyzin and N. Reyzin. Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying. In *Information Security and Privacy*, volume 2384 of *LNCS*, pages 144–153. Springer-Verlag, 2002.

[24] S. Seys and B. Preneel. Power consumption evaluation of efficient digital signature schemes for low power devices. In *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob'2005)*, pages 79–86, Aug. 2005.

[25] R. Steinwandt and V. I. Villányi. A one-time signature using run-length encoding. *Information Processing Letters*, 108(4):179 – 185, 2008.

[26] D. R. Stinson. Some observations on the theory of cryptographic hash functions. *Designs, Codes and Cryptography*, 38(2):259–277, 2006.

[27] C. Tartary, H. Wang, and S. Ling. Authentication of digital streams. *IEEE Transactions on Information Theory*, 57(9):6285–6303, Sept. 2011.

[28] UCTE. *Final Report of the Investigation Committee on the 28 September 2003 Blackout in Italy*, Apr. 2004.

[29] O. Ugus, D. Westhoff, and J.-M. Bohli. A ROM-friendly secure code update mechanism for WSNs using a stateful-verifier τ-time signature scheme. In *WiSec '09: Proceedings of the second ACM conference on Wireless network security*, pages 29–40. ACM, 2009.

[30] U.S.-Canada Power System Outage Task Force. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*, Apr. 2004.

[31] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt. Time valid one-time signature for time-critical multicast data authentication. In *IEEE INFOCOM 2009*, pages 1233 –1241, Apr. 2009.

[32] W. Wang, Y. Xu, and M. Khanna. A survey on the communication architectures in smart grid. *Computer Networks*, 55(15):3604–3629, 2011.

[33] J. Zhang and C. A. Gunter. Application-aware secure multicast for power grid communications. *International Journal of Security and Networks*, 6(1):40–52, 2011.