Real-time Digital Signatures for Named Data Networking

Charalampos Katsis
Department of Computer Science
Purdue University
ckatsis@purdue.edu

Ankush Singla
Department of Computer Science
Purdue University
asingla@purdue.edu

Elisa Bertino
Department of Computer Science
Purdue University
bertino@purdue.edu

ABSTRACT

Digital signatures are a fundamental building block for ensuring integrity and authenticity of contents delivered by the Named Data Networking (NDN) systems. However, current digital signature schemes adopted by NDN open source libraries have a high computational and communication overhead making them unsuitable for high throughput applications like video streaming and virtual reality gaming. In this poster, we propose a real-time digital signature mechanism for NDN based on the offline-online signature framework known as *Structure-free and Compact Real-time Authentication scheme* (SCRA). Our signature mechanism significantly reduces the signing and verification costs and provides different variants to optimize for the specific requirements of applications (i.e. signing overhead, verification overhead or communication cost). Our experiments results show that SCRA is a suitable framework for latency-sensitive NDN applications.

CCS CONCEPTS

• **Security and privacy → Digital signatures**; Security protocols.

KEYWORDS

Digital signatures, Named data networks, Real-time authentication, Signature aggregation, Time-critical applications

ACM Reference Format:
Charalampos Katsis, Ankush Singla, and Elisa Bertino. 2020. Real-time Digital Signatures for Named Data Networking. In *7th ACM Conference on Information-Centric Networking (ICN '20), September 29-October 1, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3405656.3420227

1 INTRODUCTION

Named Data Networking (NDN) [2, 14, 15] has emerged as an innovative network architecture that fundamentally rethinks the way content is distributed over networks. NDN allows consumers to indicate the contents they want by just providing the "name" of the content, rather than providing the content location.

An important requirement when dealing with content distribution is to assure content integrity and authenticity. Current designs of NDN systems have adopted the well known Public Key Infrastructure (PKI) for content authentication.

ECDSA [5] and RSA [7] are two signature schemes supported by several open source NDN libraries [8, 9]. A drawback of these traditional signature schemes is that they introduce a significant computational overhead, adding a lot of latency to communication. In real-time applications, like video conferencing, VR gaming etc., it is essential to keep the latency to a minimum to maintain a certain quality of service. NDN provides a temporal in-network storage caching for the recently requested data packets. However, this technique is not sufficient to reduce communication latency. It has been repeatedly shown that the aforementioned signatures schemes impose a bottleneck in the overall communication for real-time applications [3, 12, 13].

2 PROPOSED SCHEME

Structure-Free and Compact Real-time Authentication (SCRA) is a suite of real-time digital signatures schemes that provide delay-aware authentication for time-critical networks [13]. SCRA transforms any aggregate signature into a signer efficient signature. It pushes the costly signature generation operations to an offline phase and uses efficient aggregation operations in the online phase to generate the actual signatures for messages. We instantiate SCRA-C-RSA in the context of NDN as it is proven to be both signer and verifier efficient. The SCRA framework has three phases: offline, online, and verification. We assume that trust has already been established between the NDN content producer and the consumer.

SCRA-C-RSA is equivalently secure to its RSA origin and it is characterized by signature immutability. Signature immutability refers to the difficulty of computing valid aggregated signatures from a set of other aggregated signatures [6].

Offline phase: This is a one-time setup operation performed by the producer to generate a precomputed table of signatures (table Γ) that is stored in memory for subsequent use. We leverage the SHA-256 hashing algorithm to calculate the hash output of the packets which are signed by the algorithm.

The producer first choose two parameters L and d such that $d \cdot L = b$, where L is the number of chucks the hash is sliced into, d is the number of bits per chunk, and b is the number of bits of the hash output (256 bits for the SHA-256 hashing algorithm). For example, when $L = 32$ and $d = 8$, we divide the hash into 32 chunks of 8 bits each. The producer then creates a precomputed signature table Γ that contains signatures calculated on all the different bit combinations for each chunk. The chunks are indexed from $i = 1$ to L. Let j be a counter in the range $0 \leq j < 2^d$. For all i and j, the producer computes $\gamma_{i,j} \leftarrow H(i||j)^u \mod n$, where H is the one-way hash function, $i||j$ the concatenation of i and j, u is the

ICN '20, September 29-October 1, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-8040-9/20/09…$15.00
https://doi.org/10.1145/3405656.3420227

	ECDSA-256	RSA-3072	SCRA-C-RSA (3072 bit key)	
			L=32	L=16
Public key size (bytes)	91	422	422	422
Average signature size (bytes)	71	384	**5.82**	**5.82**
Average signing time (msec)	**0.059**	1.49	**0.22**	**0.11**
Average verification time (msec)	0.10	0.063	**0.040**	**0.02**
Average end-to-end delay (msec)	0.35	1.85	**0.46**	**0.30**

Table 1: Standalone message authentication with 100-packet signature aggregation.

producer's private key and n is the public modulus. Overall, the total number of computations is $L \times 2^d$.

Online phase: Once the producer constructs a data packet, it must sign it before sending it to the forwarding daemon [11]. The producer wire-encodes [10] the packet into a buffer and then hashes the encoded message and slices it into L chunks of d bits. Now, the producer simply fetches the precomputed signature from table Γ for each chunk i ($1 \leq i \leq L$) of the hashed message: $\gamma_i \leftarrow \Gamma[i][hash[i]]$, where γ_i is the cached precomputed signature in table Γ, $hash[i]$ denotes the bits of the chunk i. Subsequently, the producer computes the packet's aggregated signature $s \leftarrow \prod_{i=1}^{L} \gamma_i \mod n$.

Verification phase: The consumer just needs to have access to the installed producer's certificate (i.e. the public key) to authenticate the data packet and must be aware in advance of the values of L and b parameters in order to correctly verify the aggregated signature. The verification procedure follows similar steps as the signing and the offline phase. The data packet is wire-encoded into a buffer and then hashed. Then, the hash is sliced into L chunks. Afterwards, the chunk index is concatenated with the chunk bits and hashed again. The consumer then calculates, $x_i \leftarrow H(i||hash[i])$. Finally, the consumer uses the producer's public key, e, and checks if $s^e = \prod_{i=1}^{L} x_i \mod n$. The equality of the two sides will result in the successful authentication of the data packet.

3 REAL-TIME APPLICATION OPTIMIZATIONS

Further optimization strategies for the specific real-time applications for NDN, that build up on the proposed signature framework, can be applied.

k-packet signature aggregation: SCRA-C-RSA performs notably better as more packet signatures are aggregated together. The costly exponentiation occurring at the verification phase happens once every k received NDN data packets. On the other hand, for the already adopted schemes the verification must be executed for each received data packet.

Probabilistic signing: In this approach, a producer can randomly sign the packets periodically and save communication and computational cost instead of calculating and sending signatures for all messages.

Different sizes for parameter L: As shown by the experimental results (see Section 4), the value of L plays an important role in the performance of our scheme. The choice of L affects the size and the construction time of the Γ table as well as the signing and verification times.

Inherent parallelizability of SCRA: SCRA is highly parallelizable in all its three phases. One can leverage parallel algorithms or hardware acceleration for even better performance.

	ECDSA-256	RSA-3072	SCRA-C-RSA (3072 bit key)	
			L=32	L=16
Public key size (bytes)	91	422	422	422
Average signature size (bytes)	**71**	384	384	384
Average signing time (msec)	**0.13**	3.17	**0.50**	**0.26**
Total packets signed	7,665	7,733	7,637	7,686
Average verification time (msec)	0.30	0.12	**0.16**	**0.12**
Total packets verified	1,492	1,485	1,488	1,489

Table 2: Real-time conferencing (NDN-RTC).

4 PERFORMANCE EVALUATION

In this section, we compare the performance of SCRA-C-RSA and a against the traditional signature schemes ECDSA and RSA in two contexts. We evaluate our approach in the NDN-CXX [9] and NDN-CPP [8] open-source codebases. Our measurements have been obtained from a machine equipped with an Intel® Core™ i7-9700 (8 Cores/12MB/8T/3.0GHz to 4.8GHz/65W) and 32GB of RAM.

We report the performance trade offs of SCRA-C-RSA for two values of the parameter L; $L = 32$ and $L = 16$. For $L = 32$ the Γ table is of size 3.14 MB and the construction time is 11.48 seconds whereas for $L = 16$ is 402.65 MB and 23.2 minutes respectively. We are using 128 security strength for factoring modulus as per NIST recommendations [1].

Standalone Message Authentication: The experiment consists of 10,000 interest-data packet exchanges between consumer and producer. The algorithm aggregates the signature per 100 packets. The NDN-CXX codebase has been used for this experiment. Table 1 presents the results of this experiment.

Real-time video conferencing: NDN-RTC [4] is a video conferencing library, implemented on top of NDN-CPP [8], designed to provide low-latency real-time communication over NDN. The message authentication functionality is handled in NDN-CPP which is where our scheme is implemented. In this experiment we had a producer application running for 100 seconds and consumer application running for 50 seconds. We assume that the consumers are not data publishers. Our scheme signs every manifest packet just as the already adopted schemes do. Table 2 reports our results on NDN-RTC implementation.

5 FUTURE WORK

As future work, we plan to propose changes to the bootstrapping process and possibly add naming conventions to support SCRA-based message authentication. The communicating entities must be able to agree on the number of aggregated signatures and update this number dynamically based on the error rate, the network congestion etc. We also plan to modify the design of NDN-RTC [4], in order to efficiently support signature aggregation using SCRA. We plan to carry out a comprehensive performance analysis of our scheme and compare it to the technique used in NDN-RTC, which is based on signed manifests containing concatenated hashes of frame segments. Finally, we will be investigating the impact of signature aggregation for schemes with smaller signature sizes.

ACKNOWLEDGEMENT

The work reported in this paper has been supported by NSF grant 1719369.

REFERENCES

[1] [n.d.]. *BlueKrypt: Cryptographic Key Recommendation.* Retrieved September 1st, 2020 from https://www.keylength.com/en/4/

[2] Alex Afanasyev, Jeff Burke, Tamer Refaei, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2018. A brief introduction to Named Data Networking. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 1–6.

[3] Tohru Asami, Byambajav Namsraijav, Yoshihiko Kawahara, Kohei Sugiyama, Atsushi Tagami, Tomohiko Yagyu, Kenichi Nakamura, and Toru Hasegawa. 2015. Moderator-controlled information sharing by identity-based aggregate signatures for information centric networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. 157–166.

[4] Peter Gusev and Jeff Burke. 2015. Ndn-rtc: Real-time videoconferencing over named data networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. 117–126.

[5] Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security* 1, 1 (2001), 36–63.

[6] Einar Mykletun, Maithili Narasimha, and Gene Tsudik. 2004. Signature bouquets: Immutability for aggregated/condensed signatures. In *European Symposium on Research in Computer Security*. Springer, 160–176.

[7] Ronald L Rivest, Adi Shamir, and Leonard Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 21, 2 (1978), 120–126.

[8] NDN team. [n.d.]. *NDN-CPP: A Named Data Networking client library for C++ and C.* Retrieved September 1st, 2020 from https://github.com/named-data/ndn-cpp/blob/master/README.md

[9] NDN team. [n.d.]. *NDN-CXX: NDN C++ library with eXperimental eXtensions.* Retrieved September 1st, 2020 from http://named-data.net/doc/ndn-cxx/current/

[10] NDN team. [n.d.]. *NDN-CXX: Wire encode.* Retrieved September 1st, 2020 from http://named-data.net/doc/ndn-cxx/current/doxygen/d4/d83/classndn_1_1Data.html#a59991bec77f7f2a161c049f6efcf79df

[11] NDN team. 2018. *NFD Developer's Guide.* NDN Technical Report NDN-0021.

[12] Lijing Wang, Ilya Moiseenko, and Lixia Zhang. 2015. Ndnlive and ndntube: Live and prerecorded video streaming over ndn. *NDN, Technical Report NDN-0031* (2015).

[13] Attila Altay Yavuz, Anand Mudgerikar, Ankush Singla, Ioannis Papapanagiotou, and Elisa Bertino. 2017. Real-time digital signatures for time-critical networks. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2627–2639.

[14] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[15] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D Thornton, Diana K Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, Christos Papadopoulos, et al. 2010. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC* 157 (2010), 158.