Improving the Efficiency of Thermal Covert Channels in Multi-/many-core Systems

Zijun Long*, Xiaohang Wang*, Yingtao Jiang†, Guofeng Cui*, Li Zhang*, Terrence Mak‡§
*School of Software Engineering, South China University of Technology, Guangzhou, China
†Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA
‡School of Electronics and Computer Science, University of Southampton, UK
§Guangzhou Institute of Advanced Technology, Guangzhou, China
Emails: 201530612446@mail.scut.edu.cn, xiaohangwang@scut.edu.cn, yingtao.jiang@unlv.edu,
{201430610429,201721045909}@mail.scut.edu.cn, tmak@ecs.soton.ac.uk

***Abstract*—In many-core chips seen in mobile computing, data center, AI, and elsewhere, thermal covert channels could be established to transmit data (*e.g.*, passwords), supposedly to be kept secret and private. Effectiveness of a thermal covert channel, measured by its transmission rate and bit error rate (BER), is so much dependent on the thermal noise/interference imposed on the channel. In this paper, we present a few techniques to improve the capacity of thermal covert channel by overcoming the thermal interference. In particular, data in a thermal covert channel are encoded and represented following a new thermal signaling scheme where logic value, 0 or 1, modules the thermal signals duty cycle. Next, we show in this study that proper selection of transmission frequency can significantly minimize thermal interference. In addition, we propose a robust end-to-end communication protocol for reliable communications. Our experiments have confirmed that, compared to an existing thermal covert channel attack [1] [2], a thermal covert channel enhanced with all the improvements proposed in this study is seeing significant BER reduction (by as much as 75%), and transmission rate boost (by more than threefold). Building such a strong thermal covert channel is the key step towards developing robust defense and countermeasures against information leaking over thermal covert channel.**

I. Introduction

Of many different types of side channels that may exist in a many-core system, a thermal covert channel, which uses heat as its media to transmit information, can be particularly dangerous [1] [3]. To create a thermal covert channel, a program running at the source produces temperature signals, while another program running at the destination picks up these signals from its thermal sensor. Most modern many-core chips have incorporated multiple temperature sensors, typically made accessible through simple software tools, as an integral part of the Dynamic Thermal Management (DTM) needed [4].

This research program is supported by the Natural Science Foundation of China No. 61376024 and 61306024, Natural Science Foundation of Guangdong Province 2015A030313743, Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund, and the Science and Technology Research Grant of Guangdong Province No. 2016A010101011 and 2017A050501003, and Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (No. 2014TQ01X590), and National Training Program of Innovation and Entrepreneurship for Undergraduates (No. 201710561160).

Many studies have demonstrated that a thermal covert channel can be exploited for data ex-filtration from a protected core to an unprotected core. As shown in Fig. 1, a program at the source end will control its power consumption by running CPU-stress code, thus producing thermal signals, while another program running at the destination end will pick up the thermal signals from its thermal sensor reading, and extract digital data out of the thermal signals.

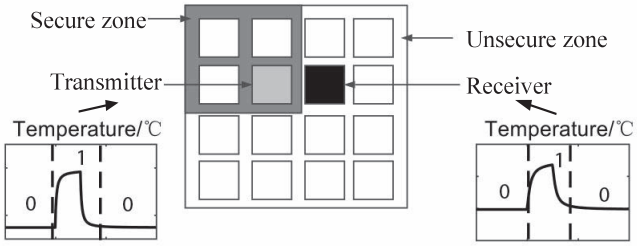

Fig. 1: Illustration of a thermal covert channel.

Yet before any strong defense line can be built, we must gain a full understanding of how capable and how robust such a thermal covert channel can be. This motives us to address many key issues that hamper the efficiency of thermal covert channel. For instance, our experimental study has revealed that a thermal covert channel is severely impacted by heat generated by other active cores. This thermal interference problem has to be tackled by exploring an observation that temperature variation caused by most normal applications (see one example in Fig. 2) tends to be very small. Fig. 2(a) shows the temperature of a core running Barnes of SPLASH-2 [5] for a time span of $150ms$, while the corresponding frequency spectrum is shown in Fig. 2(b). Although the thermal signals seen in Fig. 2 can interfere the data transmission over a thermal covert channel, as one can see that these signals are concentrated in low frequency band, and thus, any secret data that need to transmit over a thermal covert channel would be better modulated and transmitted at a higher frequency to mitigate the thermal interference.

In this paper, we attempt to address three major areas that hamper the efficiency of a thermal covert channel, namely signal generation at the source end, signal decoding at the destination end , and the end-to-end communication protocol.

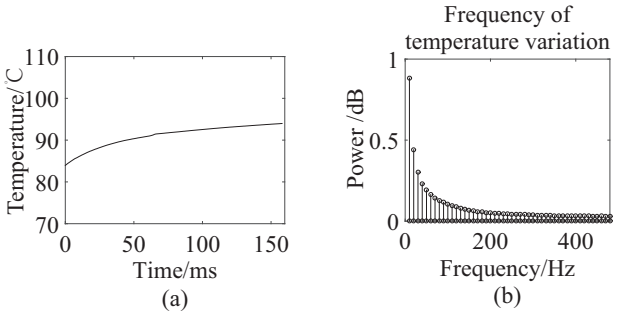

Fig. 2: (a) The temporal thermal profile of a core running Barnes, and (b) the corresponding power spectrum

Improvements at these areas shall make a thermal covert channel more reliable and less sensitive to environment.

Effects and effectiveness of a thermal covert channel are determined in terms of the transmission rate (measured in bits transmitted per second) and bit error rate (BER), which are found to be related to several system parameters. The result shows that, compared to an existing thermal covert channel attack [1], our proposed attack can reduce BER by 75%, and improves transmission throughput by as much as 370%.

The rest of the paper is organized as follows. Section II presents the background and the related work. A new thermal encoding scheme is detailed in Section III, and Section IV discusses frequency selection for transmitting thermal signals. Better performance of a thermal covert channel requires a new end-to-end protocol design, as Section V is dedicated to this purpose. Section VI reports the experimental results. Finally, Section VII concludes the paper.

II. Background and Related Work

A. Background

1) Temperature sensor implementation: We assume that each core has a thermal sensor [6], which is implied by modern many-core chips with fine-grain power budgeting [7], [8].

As Intel processors occupy 99% market in sever system and data center, and thermal covert channel is unprecedented threat to sever system and data center, we put our analysis mainly in Intel processors. Preventing damage from overheating and permanent silicon damage, Intel set each of its core with a maximum junction temperature which is the highest temperature that safe for executing. To avoid such accident, Most Intel processors equipped each core a digital thermal sensor to monitor temperature variations. Furthermore, with dynamic thermal management, processor can have a better performance. For this reason, these Intel processors all provide software interface to internal thermal sensor for a more intelligent thermal management policy. From our testing, these sensors can be easily accessible on computer system running either Windows or Linux through simple tools that export temperature information to user base space processes. These thermal data can be acquired from special CPU registers. The data from all sensors is readable through the core-temp kernel module on Linux systems. These records can extract from user space through the sysfs file system which is refreshed every $2ms$.

The accuracy of these digital thermal sensors varies from different generations of Intel processors or different brand of processors. But, typically, they have a resolution of ± 1 ^{o}C.In some latest version processors, the accuracy of digital thermal sensors can up to ± 0.1 ^{o}C.

2) Thermal behavior model: The most popular abstraction of the thermal behavior of a processor is the resistor-capacitor mesh network model. This model is based on the famous well-known duality between thermal and electrical phenomenon. The heat flow transfer between layers and by air. In this analysis, its reasonable to model a processors thermal behavior by using a linear model. It contains factors such as high thermal correlation, heat-sink and fan cooling parameters. We test and analyze all these factors impact in Hotspot simulation.

B. Thermal Covert Channels

Covert channel attacks have been studied for cloud systems [9], operating system [10], many-core chips [11]–[15] and *etc.* Covert channels can be broadly classified as storage or timing channels. The thermal covert channel attacks in many-core systems fall into the category of storage attack, where a program running in the source core (hereafter referred as transmitter) affects the temperature that another core(*i.e.*, receiver) can observe [16].

Recis *et al.* [3] studied thermal related attack where passwords are transmitted by reading the speed and period of the fan. The angular velocity of the fan changes according to the chip temperature to prevent it from overheat. Therefore, the transmitter can encode the bit streams of passwords by running the chip with different power consumptions and temperatures. The receiver then can decode the passwords by reading the angular velocity of the fan.

Masti *et al.* [1] measured the capacity of thermal covert channels that cross one hop or tow hops of multi-core processors. In [1], the transmitter encodes the bits of a data stream (*e.g.*, a password): logic '1' is represent as a thermal pulse with certain duration of higher temperature (by having the transmitter core active for that duration), while logic '0' as low temperature (by turning off the transmitter). The receiver reads temperature signal from its thermal sensor and decodes the data accordingly. So far, they have achieved a transmission rate of up to $1.33bps$ with and BER of 11% for the 1-hop channel on Intel Xeon-based sever. Bartolini *et al*. [2] extended the Masti *et al* [1] work using a different encoding scheme. Logic '1' is encoded by changing the state of core from active to sleep, and logic '0' is by changing the state of core from sleep to active.

Even though their works used both experimental result and theoretical analysis to characterize thermal covert channels that can be created in modern multi-cores, they ignored the interference caused by other active cores In the next sections, we shall show that the thermal-induced interference on thermal covert channel can be dealt with a a new encoding scheme and selection of proper transmission frequency of thermal signals.

Another issue is that these works implicitly assumed there was just one thermal channel from a source to a designation. We will show later that we can allow multiple thermal covert channels that have multiple sources, but one destination at the same time.

III. New Duty-Cycle-Based Encoding Scheme

A. Single-bit Single-channel

We design a new encoding scheme, character as single-bit single-channel, which is the basis form of thermal covert channel. In Single-bit Single-channel, we only use one frequency in a thermal covert channel at one time. The encoding scheme is illustrated in Fig. 3. Logic '1' is encoded by first high temperature and low temperature in 50% duty cycle. Logic '0' is encoded by keep low temperature in one period. The power consumptions generating the two temperature variations.

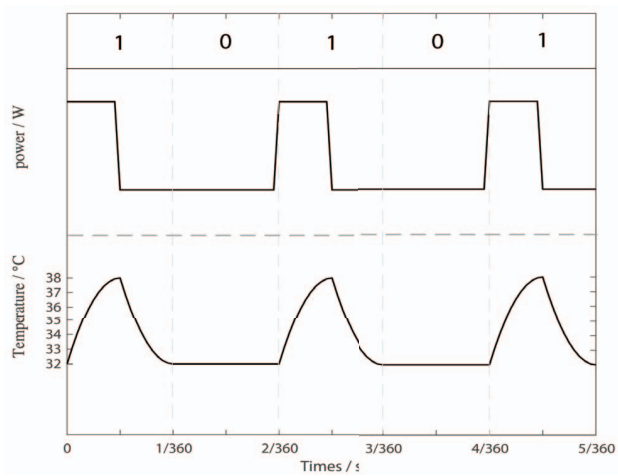

Fig. 3: Encoding scheme of logic '1' and logic '0'.

And next, the receiver uses the following rule to decode the bit streams. If the amplitude of a certain frequency component is higher than a preset threshold, it is decoded as 1, otherwise it is 0. The threshold of the amplitude of the frequency components in Fig. 3 are set empirically. We observed that in Fig. 4, by setting the amplitude to be higher than $0.05dB$, the data in the covert channel can be decoded without interruption.

B. New character of thermal covert channel: Single-bit Multi-channel (Multiple transmitters to one receiver)

In the same token, if there is more than one thermal covert channel, inter channel interference can be minimized by selecting different frequencies for different channels, and all these signal frequencies must be set again high enough to mitigate the thermal interference from other cores.

To support multiple thermal covert channel communication between multiple transmitter and one receiver, each thermal covert channels can be separated by different frequencies, as in Fig. 5. In Fig. 5, there are two covert channels, A and B, initialed by core A and core B in the chip, send message to same receiver at the same time. Their temperature variation frequencies are set to be different by controlling the lengths of periods of their time intervals T_A and T_B. Therefore, the two covert channels work without interfering each other.

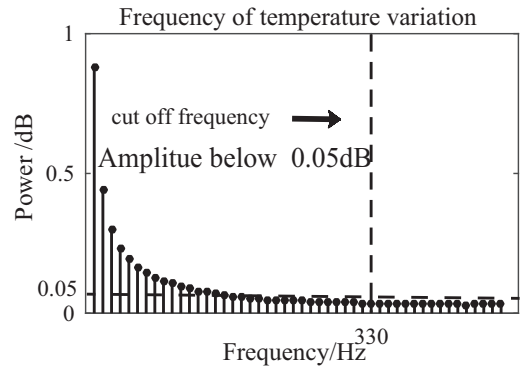

Fig. 4: The amplitudes of frequency components over $330Hz$ are lower than $0.05dB$, for active cores running normal applications. Therefore, the base amplitude of the thermal signal is set to be over $0.05dB$.

IV. Transmission Frequency Selection

Past studies have shown that thermal signals in a thermal covert channel could be corrupted or even completely jammed by the massive amount of heat generated by other active cores in the chip. In this study, this critical transmission issue is tackled by exploring the observation that temperature of cores running most normal applications tends to be quite stable or vary only at a very low rate.

To reduce interference from other active core, we set the frequency of our temperature signal higher than that of cores running normal applications. As shown in Fig. 2, the temperature of cores running normal applications is changing at low frequency. As a result, by selecting frequency of thermal signals high enough, thermal interference coming from active cores can be considerably suppressed. In our work, the base frequency of the covert channel is set to be over $330Hz$ so that it is not interfered by noise, as shown in Fig. 4.

Also, due to the physical effects, and the heat transfer characterizes, if we assume the power supply up to $200W$ and cooling system strong enough (like air force convection between 20 and $200W/(m^2\dot{K})$, it possible for us to produce the frequency of thermal signal more than $330hz$ but not higher $600hz$ in our practice.

V. Communication Protocol Design

A. Notification Flow

For more reliable thermal covert channel, a notification flow is used to initiate and terminate a transmission session (Fig.6). We preset $340hz$ as notification channel. All notification flow encoded by thermal signal should use $340hz$ as frequency. If more than one thermal covert channel need to communication, add one each step, like $341hz$, $342hz$,and 20 thermal covert channel exist one time at most. A transmission starts by an REQ packet (111) from the transmitter to the receiver.

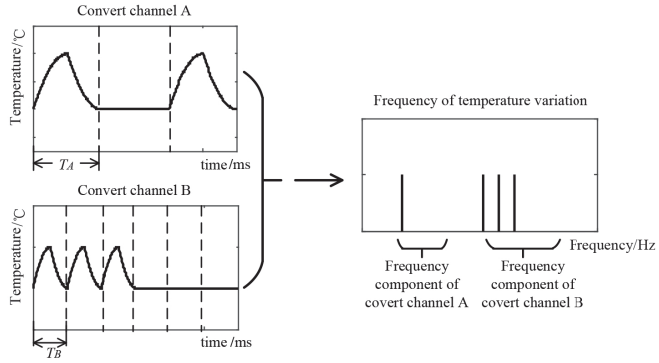

Fig. 5: Supporting multiple thermal covert channels. Multiple thermal covert channels can send information at the same time, and receiver can decode information from different frequency component in the frequency domain.

The transmitter keeps waiting until it receives an ACK1(001) coming back from the receiver. Then transmitter sends the second ACK ACK2(1101) followed by the actual data payload to the receiver. Once all the data are transmitted, a specific END (101010) packet is sent to the receiver to terminate the transmission.

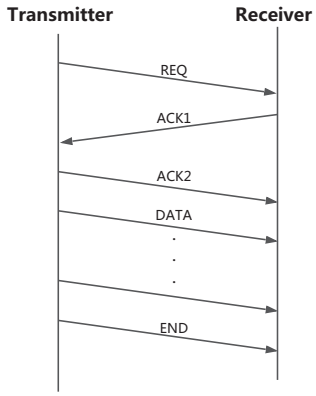

Fig. 6: The communication protocol of the covert channel.

B. Data flow

In further study, we choose $360hz$ as priority frequency to use by empirical. If multiple thermal channels require more frequencies, we add 10step each time, like $370hz$, $380hz$, $390hz$. And apply them to each thermal covert channel, respectively. In this selection, not only can we support high transmission rate in thermal covert channel but also easy to produce responding temperature variations, because they do not change that fast.

When transmitting data flow like sensitive data or password, we encode them into thermal signal as encoding scheme state above. During the transmission, we have a short stop for one second each five second. This arrangement let transmitters temperature drop and restore to original level, which more easy and arcuate for encoding next time. With this recovery, BER will drop significantly.

Many core configuration	
Number of processors	16/36/64(MIPS ISA 32 compatible)
Fetch/Decode/Commit size	4/4/4
ROB size	64
L1 D cache(private)	16 KB, two-way, 32 B line, two cycles, two ports, dual tags
L1 i cache(private)	32 KB, two-way, 64B line, two cycles
L2 cache(shared) MESI	64 KB slice/node, 64 B line
Protocol	six cycles, two ports
Main memory size	2 GB, latency 200 cycles
Benchmarks	
PARSEC	streamcluster, swaptions, ferret, blackscholes freqmine, dedup, canneal, vips, fluidanimate
SPASH-2	barnes, raytrace
AES	encoder, decoder

TABLE I: Configuration used in the simulation

VI. Experimental evaluation And Discussion

A. Experimental Setup

Experiments were performed on an even-driven C++-based many-core simulator [17], with DSENT integrated as the power model and Hotspot [18] as the temperature simulator. In this simulation, multiple applications can arrive at the system simultaneously. The experiments were carried out on various many-core systems, with with 4×4, 6×6 and 8×8 cores. Table I lists the simulator configuration. We used three kinds of benchmarks in the experiment, which are also listed in Table I. The benchmarks are selected from PARSEC [19], SPLASH-2 [5], and AES [20]. In each experiment, cores other than the transmitters and receivers are running the threads of these benchmarks. Each benchmark is parallelized into4, or 8, or 16 threads. The floorplan of the processor cores in [21] is adopted.

The famous thermal application HotSpota modeling methodology for developing compact thermal models based on the popular stacked-layer packaging scheme in modern very large-scale integration systems. The HotSpot compact thermal modeling approach is especially well suited for preregister transfer level (RTL) and presynthesis thermal analysis and is able to provide detailed static and transient temperature information across the die and the package.

The HotSpot modeling approach was first validated through detailed FEMs in FloWorks [18]. It was also validated by comparing with real temperature measurements from a commercial thermal testing chip [22]. Regardless of the relatively cool die temperature, multiple paper and experiment confirm that the errors between the HotSpot model and the thermal sensor measurements are within 10% of the measured temperatures.

In our designed thermal covert channel that supports end-to-end communications between a source (transmitter) and a destination (receiver) pair, it is imperative to represent the data as thermal signals, and use a reliable communication protocol.

B. Source and Destination end program

In this part, we demonstrate some core code of our program at source end and receiver in Fig. 7.At the source end of a thermal covert channel, bit streams of secret data are best encoded and represented as thermal signals (*i.e.*, temperature variations) with varying duty cycles.

```
int Dutycycle, Bit;
while(!cin.eof()){
cin>>Dutycycle;
cin>>Bit;
if(Bit)
Encode_1(Dutycycle);
//Stress CPU
else
Encode_0(Dutycycle);
//Idle CPU
}
```

(a)

```
string Receiver;
ofstream Data;
//Initialize
while(!interrupt){
Record(&Receiver);
//Sameple every T
}
Data<<Receiver;
```

(b)

Fig. 7: (a) Source ends program.And the source program transmitter affects the duty cycle of the power consumption of the core by executing a heavy loop like one of popular CPU-burn stress-test, and (b) Destination ends program.And the program at the receiver only records a temperature trace by reading the sensors and later sends it to the attacker through the network. Both programs can run at the user level in popular operating system like Linux or Windows, Android.

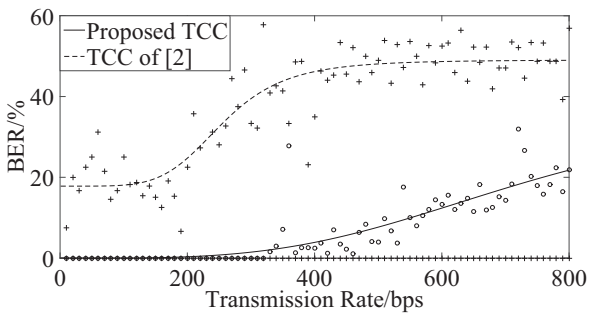

Fig. 8: Comparison of the BERs of two thermal covert channels(TCC).

C. Evaluation the efficiency of the proposed thermal covert channel

Fig. 8 compares the attack effect of the proposed thermal cover channel with that of [2]. The system size is 8×8, the number of thermal cover channel is 2 and the average distance between transmitter and receiver is 1 hop. From Fig. 8, one can see that, when the transmission rate is high, *e.g.*, $800bps$, the BER of our proposed channel is $0.4x$ of that in [2]. On average, the BER of our proposed channel is $0.25x$ of that in [2]. In contrast, the thermal cover channel in [2] suffers more interruption from noise.

Fig. 9 and Table II compare the BER of the proposed thermal covert channel with that of [2] under different system sizes. The number of thermal cover channel is 2, and the average distance between transmitter and receiver is 1 hop.

System size	BER of the Proposed TCC	BER of the TCC in [2]
4×4	35.52%	40.31%
6×6	24.88%	41.59%
8×8	23.35%	40.73%

TABLE II: Comparison at different system sizes.

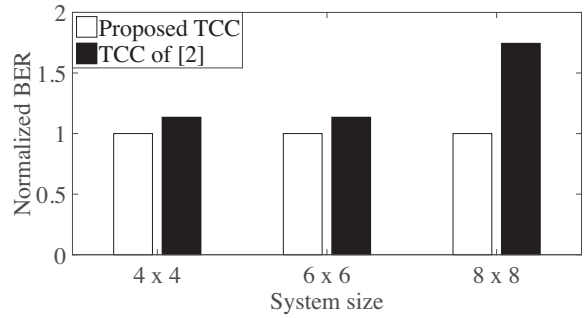

Fig. 9: Comparison at different system sizes.

Number of covert channels	Transmission rate of Proposed TCC [bps]	Transmission rate of TCC of [2] [bps]
2	40	32
4	80	61
6	120	91
8	160	119
10	200	146
12	240	174
14	280	191
16	302	203

TABLE III: Comparison at different numbers of thermal covert channels.

From Fig. 9, one can see that, when the system size is 8×8, the BER of our proposed thermal cover channel is only $0.57x$ of that in [2]. On average, the BER of our proposed thermal cover channel is 0.68 of that in [2].

Fig. 10 and Table III compare the transmission rate of the proposed thermal covert channel with that of [2] under different numbers of thermal cover channel in the same chip. The system size is 8×8, and average distance between transmitter and receiver is 1 hop. From Fig. 11, one can see that, when the number of covert channels is high, *e.g.*, 16, the transmission rate of our proposed thermal covert channel is 50% better than that in [2]. On average, the transmission rate of our proposed thermal covert channel is 30% better than that in [2]. The covert channels in [2] interfere each other as the frequencies of the thermal signals of different covert channels are not complete different. Our proposed covert channel assigns different frequencies of thermal signal for each covert channel transmission and avoids interference.

Fig.11 and Table IV compare the transmission rates of the proposed thermal cover channel against those of [2] under different average distances between transmitter and receiver. The system size is 8×8, and the number of thermal covert channel is 3. From Fig. 11, one can see that, when the average distance is long, *e.g.*, 3 hops, the transmission rate of our proposed thermal cover channel is 38% higher than that of [2]. On average, the transmission rate of our proposed thermal covert channel is 28% higher than that in [2].

Distance (hop counts)	Transmission rate of Proposed TCC [bps]	Transmission rate of TCC of [2] [bps]
1	20.0	16.5
2	20.0	16.0
3	20.0	14.5

TABLE IV: Comparison at different average distances between transmitters and receivers.

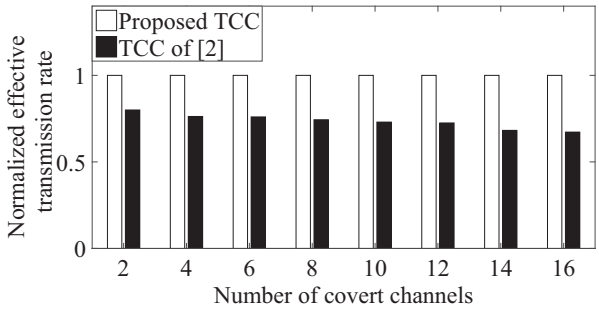

Fig. 10: Comparison at different numbers of thermal covert channels.

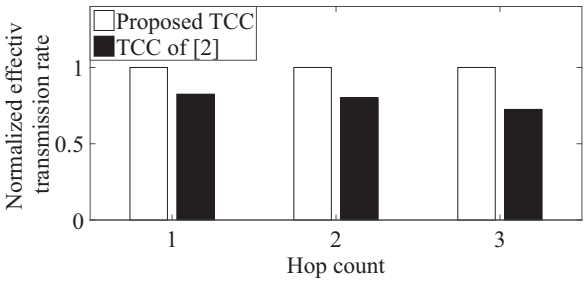

Fig. 11: Comparison at different average distances between transmitters and receivers.

In summary, the proposed thermal cover channel has higher transmission rate and lower BER than thermal cover channel in [2] under many environments.

VII. Conclusion

This paper addressed a few important issues pertaining to the efficiency improvement of thermal covert channel. At the source end of a thermal covert channel, data are encoded and represented following a new signaling scheme that explores temperature variations and allows signals to vary their duty cycles. Since thermal signals in a thermal covert channel could be easily corrupted or even completely jammed by massive amount of heat generated by other active cores in the chip, we showed in this study that proper selection of transmission frequency can significantly minimize thermal interferences. Besides the improvements made at the transmitter, the channel, we also proposed a robust end-to-end communication protocol for reliable communications. With our improvements, experimental results have confirmed that, compared to an existing thermal covert channel attack, our proposed attack can reduce BER by 75%, and improves transmission rate 370%. Therefore, the proposed attack can be used to transmit password secretly in a chip with a stable transmission rate of 160bps with a BER as low as 0%.

References

[1] R. J. Masti, D. Rai, A. Ranganathan, C. Müller, L. Thiele, and S. Capkun, "Thermal covert channels on multi-core platforms," in *Proc. USENIX Security Symp.*, 2015, pp. 865–880.

[2] D. B. Bartolini, P. Miedl, and L. Thiele, "On the capacity of thermal covert channels in multicores," in *Proc. ACM Conf. Computer Systems*, 2016, pp. 24–39.

[3] C. Reis, A. Barth, and C. Pizano, "Browser security: lessons from google chrome," *Queue, Distributed Computing*, vol. 7, no. 5, pp. 3–12, 2009.

[4] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. IEEE Int'l Symp. High-Performance Computer Architecture,*, 2001, pp. 171–182.

[5] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: characterization and methodological considerations," in *Proc. IEEE Int'l Symp. Computer Architecture*, 1995, pp. 24–36.

[6] S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, S. Kottapalli, and S. Vora, "A 45 nm 8-core enterprise Xeon processor," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 7–14, 2010.

[7] X. Wang, B. Zhao, T. Mak, M. Yang, Y. Jiang, and M. Daneshtalab, "On fine-grained runtime power budgeting for Networks-on-Chip systems," *IEEE Trans. Computers*, vol. 65, no. 9, pp. 2780–2793, 2016.

[8] J. Long, S. O. Memik, G. Memik, and R. Mukherjee, "Thermal monitoring mechanisms for chip multiprocessors," *ACM Trans. Architecture and Code Optimization*, vol. 5, no. 2, pp. 9–41, 2008.

[9] Z. Wu, Z. Xu, and H. Wang, "Whispers in the hyper-space: high-speed covert channel attacks in the cloud," *Proc. Symp. Usenix Security*, 2013.

[10] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of l2 cache covert channels in virtualized environments," in *Proc. ACM Cloud computing Security workshop*, 2011, pp. 29–40.

[11] Z. Wang and R. B. Lee, "Covert and side channels due to processor architecture," in *Proc. IEEE Conf. Computer Security Applications*, 2006, pp. 473–482.

[12] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," in *Proc. ACM Conf. Computer and Communications Security*, 2012, pp. 305–316.

[13] Y. Wang and G. E. Suh, "Efficient timing channel protection for on-chip networks," in *Proc. IEEE/ACM Int'l Symp. Networks on Chip*, 2012, pp. 142–151.

[14] G. Venkataramani, J. Chen, and M. Doroslovacki, "Detecting hardware covert timing channels," *IEEE Micro*, vol. 36, no. 5, pp. 17–27, 2016.

[15] P. Gu, D. Stow, R. Barnes, E. Kursun, and Y. Xie, "Thermal-aware 3D design for side-channel information leakage," in *Proc. IEEE Int'l Conf. Computer Design*, 2016, pp. 520–527.

[16] S. J. Murdoch, "Hot or not: revealing hidden services by their clock skew," in *Proc. ACM Conf. Computer and Communications Security*, 2006, pp. 27–36.

[17] X. Wang, M. Yang, Y. Jiang, P. Liu, M. Daneshtalab, M. Palesi, and T. Mak, "On self-tuning networks-on-chip for dynamic network-flow dominance adaptation," *ACM Trans. Embedded Computing Systems*, vol. 13, no. 2s, pp. 73–80, 2014.

[18] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. IEEE Symp. Computer Architecture*, 2003, pp. 2–13.

[19] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: characterization and architectural implications," in *Proc. ACM Int'l Conf. Parallel Architectures and Compilation Techniques*, 2008, pp. 72–81.

[20] Y. S. Yang, J. H. Bahn, S. E. Lee, and N. Bagherzadeh, "Parallel and pipeline processing for block ciphe algorithms on a network-on-chip," in *Proc. IEEE Int'l Conf. Information Technology: New Generations*. IEEE, 2009, pp. 849–854.

[21] X.-H. Wang, P. Liu, M. Yang, M. Palesi, Y.-T. Jiang, and M. C. Huang, "Energy efficient run-time incremental mapping for 3-d networks-on-chip," *J. Computer Science and Technology*, vol. 28, no. 1, pp. 54–71, 2013.

[22] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, "Compact thermal modeling for temperature-aware design," in *Proceedings of the 41st annual Design Automation Conference*. ACM, 2004, pp. 878–883.