frontiers | Frontiers in Energy Research

METHODS
published: 04 July 2022
doi: 10.3389/fenrg.2022.842938

Cyber Security Protection of Power System Equipment Based on Chip-Level Trusted Computing

Wei Xi, Xiaobo Li, Qihui Feng, Hao Yao, Tiantian Cai and Yang Yu*

Digital Grid Research Institute, China Southern Power Grid, Guangzhou, China

This article proposes a network security protection scheme for power system embedded devices based on chip-level trusted computing, gives the overall architecture of power equipment chip-level trusted computing, introduces the principle of trust chain transfer within power terminals, and proposes the method of using the on-chip SRAM inside the main control chip of power system embedded devices as the PUF to construct the trusted root. Furthermore, we propose a complete trust transfer mechanism from booting trust to the loading that trust on operating system and application, and give the corresponding program design framework and program logic. The chip-level trusted computing program designed in this article was applied in a distribution automation DTU device equipped with the FUXI chip that is developed by the China Southern Power Grid and the Sylix OS operating system. Finally, we carry out the security protection test. The validity of the designed program is verified by running malicious programs to attack the power equipment. The results show that the designed program can effectively detect and intercept illegal programs, and provide an effective solution for the security protection of power equipment in the open network environment.

Keywords: power equipment, cyber security, trusted computing, embedded system, on-chip security

OPEN ACCESS

Edited by:
Xiangjun Zeng,
Changsha University of Science and Technology, China

Reviewed by:
Sheng Su,
Changsha University of Science and Technology, China
Xuan Liu,
Hunan University, China

****Correspondence:***
Wei Xi
csgxiwei@foxmail.com

Specialty section:
This article was submitted to Smart Grids, a section of the journal Frontiers in Energy Research

Received: *24 December 2021*
Accepted: *09 May 2022*
Published: *04 July 2022*

Citation:
Xi W, Li X, Feng Q, Yao H, Cai T and Yu Y (2022) Cyber Security Protection of Power System Equipment Based on Chip-Level Trusted Computing. Front. Energy Res. 10:842938. doi: 10.3389/fenrg.2022.842938

1 INTRODUCTION

With the development of smart grids, digitization, openness, and interconnection have become the future trend of power grids (Srinivas et al., 2021; Ciavarella et al., 2016; Liu R. et al., 2015). It is foreseeable that a large number of terminal devices and multiple users will be connected to the power grid in the future (Bedi et al., 2018; Park et al., 2019), gradually forming an open and interactive network environment, and the information security of power system terminals, such as electric energy meters, concentrators, and protective measurement and control devices, will face unprecedented challenges. "Stuxnet virus attack" (Li, 2019), "Ukraine blackout" (Guo et al., 2016), "U.S. oil pipeline WannaCry blackmail attack" (Yu and Guan, 2021), and other incidents show that the energy system has always been the key target and object of various malicious attacks. At the same time, the bottom layer of the power system terminal is chip processor, which may also be maliciously exploited because of the vulnerability. In 2018, two vulnerabilities of Meltdown and Spectre were discovered (Efe and Güngör, 2019), which threaten hundreds of millions of PCs, smartphones, and other devices worldwide, as well as a large number of users' privacy data are at the risk of leakage (Kolias et al., 2017).

The abovementioned threats are mainly attacked by exploiting the security vulnerabilities on the computing platform, and the root cause is the lack of the active defense mechanism on the computing platform itself. In order to protect the industrial control terminals of the power system from

cyberattacks, the China power grid has established a comprehensive network security in-depth protection system based on network isolation and border protection (National Development and Reform Commission of China, 2014). In 2015, the National Energy Administration of China issued the "Overall Security Protection Plan for Power Monitoring System," requiring to "gradually promote the application of trusted computing technology," to realize the transformation of power terminals from passive defense to active immunity (Zhang, 2019).

Trusted computing is an active defense technology. It uses hardware attributes as the root of trust and measures layer by layer when the system starts to establish an isolated execution operating environment, and guarantees the security of sensitive operations on the computing platform, thus achieving the protection of a trusted code (Avizienis et al., 2004). Trusted computing establishes trust anchors by integrating dedicated hardware modules in the computing platform and uses cryptographic mechanisms to establish a trust chain, to build a trustworthy computing environment, making it possible to fundamentally solve the security problem of the computing platform (Feng et al., 2011).

Beginning late 1990s, IBM, HP, and other computer companies began to propose trusted computing technology solutions and cosponsored the establishment of the Trusted Computing Group (TCG) (Trusted Computing Group, 2003), whose purpose and goal is to ensure computing security through the development and opening of trusted computing industry standards. The TCG has published a series of specifications such as trusted platform module (TPM) (Trusted Computing Group, 2017a), trusted software stack (TSS) (Trusted Computing Group, 2017b), trusted network connection (TNC) (Trusted Computing Group, 2017c), and mobile trusted module (MTM) (Kylanpaa and Ekberg, 2007). The Institute of Software Research of the Chinese Academy of Sciences has proposed a trusted computing architecture based on the trusted cryptography module (TCM) (Feng et al., 2020) and researched the application of this system for security protection of computer systems, mobile communication terminals, and the Internet of Things, which can promote the active immunity function of the power terminal part.

Although some achievements have been made in the field of Trusted Computing, the current achievements are mainly applied to personal computers and servers. For the realization of Trusted Computing of power grid terminals, related scholars have made some explorations such as these literatures works. Zhang et al. (2017), Wang (2018), and Zheng (2019) carried out research on information security of power consumption information collection terminals based on trusted computing. Xu (2014) carried out the research and implementation of embedded power distribution terminals based on trusted computing. Sun et al. (2014) designed a trusted authentication mechanism based on trusted computing theory and trusted security chip technology, and the literature. Zhang and Zhao (2019) made a research on the trusted computing technology in power collection terminals. To sum up, in the field of power systems, the application of trusted computing technology has sprouted, but it is far from mature, and the research on the security protection of power system terminals based on trusted computing is still in the exploratory stage.

In this article, we are trying to meet the requirement; that is, the power terminals realize active security defense in the open environment, and we design a chip-level trusted computing architecture for power embedded terminals. On this basis, based on the TCM trusted computing architecture, we use the hardware properties of the chips of power system devices to build a root of trust, and establish a trusted execution environment free from malicious code attacks to ensure system entities behave as expected. The structure of this article is as follows.

2 OVERALL ARCHITECTURE OF CHIP-LEVEL TRUSTED COMPUTING FOR POWER EQUIPMENT

Damaging critical components of the computer, tampering with the running code of the system, and modifying the configuration of the system are important means of hacker attacks. These attacks actually tamper with the trusted execution environment of the original system by modifying the correct running code of the system and changing the important configuration files of the system, and then use this untrusted execution environment to achieve their own attack goals. Therefore, building a trusted computing environment for computer systems has become an urgent problem in the field of computer security.

The TCG defines trust as follows: if an entity's behavior always moves toward an expected goal in an expected manner, then the entity is credible. Based on its definition of trust, the TCG gives a way to build a trusted execution environment for computer systems: trust chain. Through the layer-by-layer measurement and trust chain transfer of the computer system, the trust from the lowest level of trusted hardware to the target application is established. Then by using the security chip that is implanted in the hardware platform to protect the measurement data, the credibility of the computer system execution environment is established.

Based on the trust chain construction technology, this article describes an embedded trusted computing security protection architecture for power industrial control terminals that includes trusted boot, operating system measurement, and application program measurement, as shown in **Figure 1**. The architecture is a set of security systems that run through the hardware layer, kernel layer, and application layer. In the system, the hardware layer uses hardware cryptographic algorithm modules and secure boot modules to enhance the computer system security; the kernel layer includes measurement modules, management modules, integrity management modules, and management and control modules, and it provides core trusted computing functions; and the application layer contains the measurement agent, kernel interface library, trusted software library, and client interface GUI, and it provides operable, manageable, and visual user application support.

The power equipment trusted computing system is responsible for collecting data and information on the

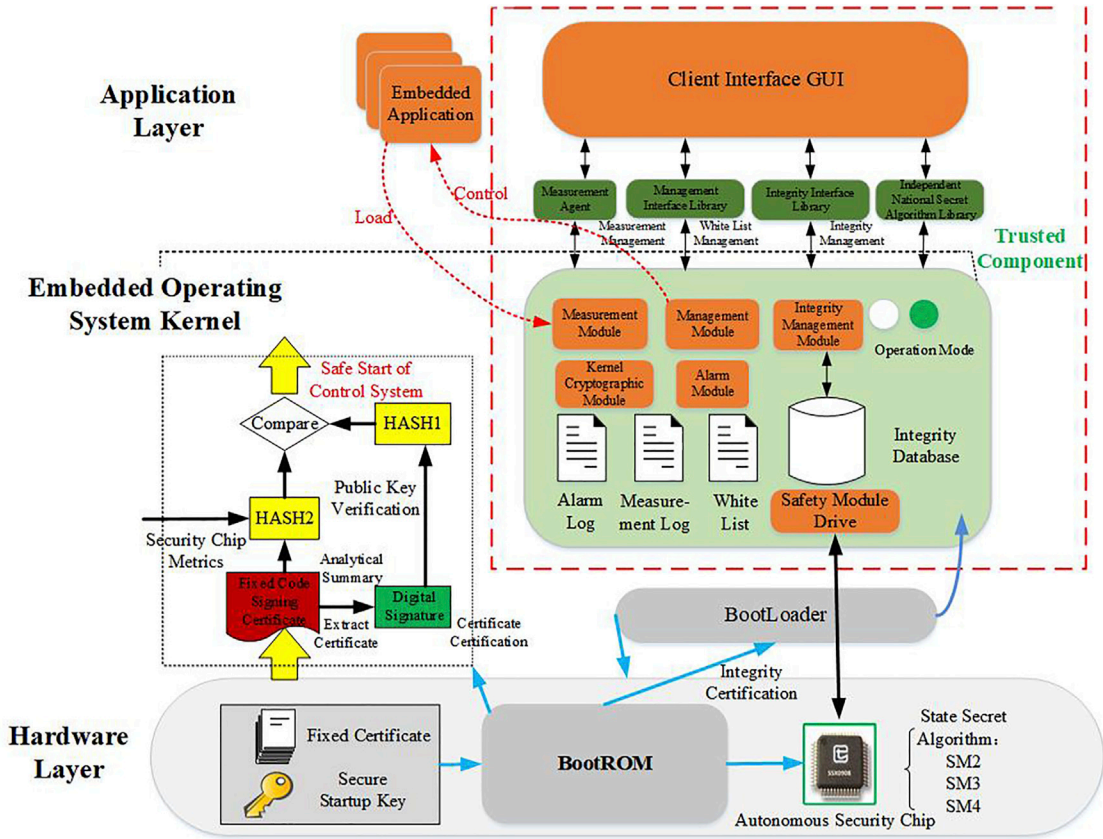

FIGURE 1 | Embedded trusted computing security protection system architecture for power industrial control terminals.

environmental integrity of the power equipment, comparing the integrity information of the equipment with the whitelist deployed on the equipment, and displaying the verification results. The administrator can make a whitelist for different devices according to specific needs. The system executes a mandatory protection function, the host enters the control state, and processes that are not on the whitelist are prohibited from running. The hardware layer of the power equipment adopts a security chip that implements a cryptographic algorithm to form the secure hardware root of a trusted computing environment.

3 THE PRINCIPLE OF TRUST CHAIN TRANSFER

The chain of trust technology starts from the root of trust, establishing a chain of trust from the underlying hardware to the application layer through gradual measurement and verification. It transfers trust from the root of trust to the uppermost application code to ensure the credibility of the entire system platform. When the system is powered on and started, each entity that obtains system control must be measured before it is allowed to run on the system. A typical trust chain construction needs to solve two problems: (1) select the entity that can be the starting point of the trust chain as the root of trust. The root of trust is the first entity in the chain of trust and the anchor of trust for the entire device, which must be safe and reliable, and (2) choose an algorithm that measures system entities to transfer trust.

This article designs a chip-level complete chain for the power system terminal equipment. The trust chain requires the hardware layer to provide components, including secure boot BootROM, secure boot keys, firmware certificates, and hardware security module.

(1) Secure boot BootROM: it is equipped with the boot code with secure boot and is solidified on the device. It is responsible for implementing security functions such as metrics and authentication firmware.
(2) Secure boot key: an asymmetric key used to sign the firmware of the device, and the public key part is solidified on the device during production.
(3) Firmware certificate: the certificate issued by the device vendor for the device firmware using the private key of the secure boot key, which can prove the legitimacy of the firmware.
(4) Hardware security module: it should provide SM2, SM3, SM4, and other independent domestic cryptographic algorithm systems, and key storage functions to provide

high-speed and secure cryptographic services for the upper-level software (Office of State Commercial Cipher Administration, 2006; Office of State Commercial Cipher Administration, 2010; Office of State Commercial Cipher Administration, 2012).

Among them, the hardware security module equipped with national cryptographic algorithm is the root of trust of the system. After booting up, the root of trust measures the boot code stored in the Bootloader, and then the system performs hash integrity calculation for each component (hardware and software module) that will be loaded during the boot process, such as OS loader, OS kernel, device driver module, initialization process, application, and network service, to obtain its metric value before it is loaded and run. The system determines the transmission of the trust chain according to the measurement value, and records the whole trust chain delivery through the measurement value, preventing the firmware loaded by the device from being tampered with, and ensuring the credibility of the operating environment when the system starts (Feng et al., 2015).

The process of trust chain execution is as follows:

(1) After the device is powered on, BootROM loads and measures the operating system loader BootLoader to obtain the measured value.
(2) Using the secure boot key on the device to verify the firmware certificate of the BootLoader firmware and verifying whether the measurement value is consistent with the standard integrity value in the firmware certificate. Only when they are consistent, the boot is allowed.
(3) BootLoader loads and metrics the OS image after booting by using the secure boot key to verify the operating system image certificate. Meanwhile, matching the measured value with the standard value in the certificate only starts the operating system kernel when it matches. The trust established by the autonomous security chip is passed to the operating system itself when the operating system is started.
(4) The operating system kernel-level extended measurement module measures all the applications loaded in the system, and guarantees the degree of trust chain with the help of independent security chips.

At each link of the trust chain transfer, the corresponding subsystems will be built, such as the trusted boot system (TCWG GRUB), the operating system measurement architecture (TCWG OSMA), the TCWG dynamic measurement, and the component measurement system. During the initialization phase of the operating system, the kernel measurement module supporting the SM3 algorithm will be automatically loaded to measure the executable program during the operating system boot process. The images of all the executable programs are first loaded into the system memory, and the measurement module analyzes the images loaded in the memory and calculates the integrity hash function, and expands the calculation results to the internal storage of the autonomous security chip for safe storage.

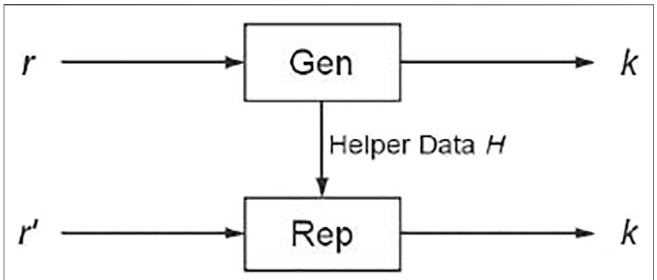

FIGURE 2 | Logic realization of the PUF.

4 REALIZATION OF CHIP-LEVEL TRUSTED COMPUTING

4.1 Construction of the Root of Trust

The physical unclonable function (PUF) (Liu Z. L. et al., 2015) refers to the use of random process deviations of integrated circuit chips in the manufacturing process to cause differences in the device size or electrical characteristics between chips, and using the readout circuits to sample the random characteristics to obtain a unique and unclonable data.

The PUF circuit is derived from the random deviation process of the analog circuit structure, and since the process deviation of each circuit is different, the difference is completely random. The PUF circuit is unique and cannot be cloned. The PUF value is output by the PUF circuit when it is powered on, and the value does not exist in non-volatile memory such as EEPROM or FLASH. It has the characteristics of being lost when power is off and cannot be read, which greatly improves the non-clonable characteristics of the PUF value (Mispan et al., 2015; Song, 2019).

The realization of the PUF includes two parts, namely, production (Gen) and reconstruction (Rep). The Gen algorithm extracts key k from PUF response r, and generates auxiliary data H, which does not include confidential data and does not require secure storage. The Rep algorithm takes H as a parameter to regenerate key k from PUF's noisy response r'. Key k is generated by the PUF owner or issuer in the Gen algorithm, so it is easy to bind new key k' to the PUF by re-running the Gen algorithm and obtaining new auxiliary data H' (3rd Generation Partnership Project (3GPP), 2018a; 3rd Generation Partnership Project (3GPP), 2018b).

This feature enables the implementation of the key update mechanism to become possible. The logic realization of the PUF is shown in **Figure 2**.

This article uses the on-chip SRAM that is inside the main control chip of the power system embedded device as the PUF to construct the root of trust for the device. It mainly includes two stages, including the production process and the reconstruction process. The manufacturer issues the key for the device during the production process, and the device is reconstructed. The process uses its own SRAM PUF to calculate the key issued by the manufacturer in real time. The specific process is shown in **Figure 3**.

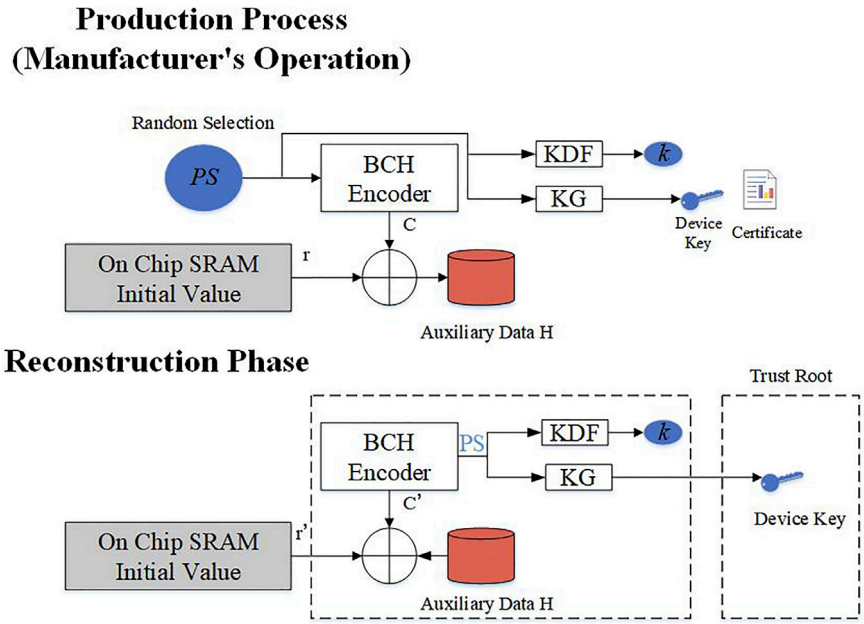

FIGURE 3 | Specific process of production and reconstruction.

4.1.1 Production Process

The production process is invoked by the vendor when the device is shipped, and it receives as input the power-up initial value of the on-chip SRAM (a binary string consisting of the initial values of the SRAM cells), and then performs the following steps.

(1) The receiving vendor randomly generates a random value *PS*, and then encodes the *PS* using a BCH error correction code to obtain a code word $C = \mathrm{BCH}_{\mathrm{Enc}}(PS)$.
(2) XOR the code word C and r, and the resulting value is recorded as auxiliary data H. H does not need to be stored safely, as long as it is stored on the persistent storage device of the device. H will help BB to regenerate *PS* during the reconstruction phase.
(3) Input a key derivation function KDF and a deterministic key generation algorithm KG to *PS*, respectively, and then generate symmetric key k and a public–private key pair (pk, sk). The manufacturer uses its signature key to issue a certificate $Cert_{pk}$ to pk. The certificate contains the standard integrity values of the embedded firmware, and these integrity values can be used to establish a chain of trust for this device.
(4) Finally, the manufacturer stores auxiliary data H and certificates $Cert_{pk}$ persistently on the device.

The main task of this stage is that the manufacturer implicitly binds the master seed *PS* to the device and issues a certificate for the device key exported by the *PS*. Here, "implicit" means *PS* that is not physically stored on the device but is calculated at run-time.

4.1.2 The Reconstruction Process of Building Blocks

After the embedded device is powered on, it first runs the code stored on BootROM, which is responsible for measuring the initial value r' of the on-chip SRAM and passing r r' to BB. SRAM is a kind of PUF, and its initial value after power-on has a certain amount of error with the manufacturer's metric value r in the production stage, which is also the reason for using error correction codes. The reconstruction process receives r' as an input, performs an XOR operation with auxiliary data H, and obtains a BCH code word with a certain error, and then the BCH decoder eliminates the error of C to obtain main seed *PS* that is bound to the device, during the production process. Finally, *PS* is used to generate symmetric key k and the asymmetric key pair of the device (pk, sk).

4.2 Transmission of Trust

The trust chain technology starts from the static trust metric root and establishes a trust chain from the underlying hardware to the application layer through step-by-step measurement and verification, passing trust from the trust metric root to the uppermost application codes to guarantee the trust of the whole system platform. The establishment of a static trust chain mainly includes two aspects: integrity measurement and trust transfer.

Trusted computing technology refers to the measurement process of a trusted entity to another entity as a measurement event. The measurement events involve two types of data: 1) the measured data, which is the expression of the code or data being measured, and 2) the measurement summary, which is the hash value of the measured data. The entity responsible for the measurement obtains the measurement summary by hashing the measured data. The measurement summary is equivalent to a snapshot of the measured data and is the integrity mark of the measured data. The measurement summary marks the integrity information of the measured data, and the integrity report needs

to use the measurement summary. Therefore, the measurement summary needs to be protected, and it is usually protected by the trusted storage root of the security chip. The measured data do not need to be protected by a trusted chip, but they need to be re-measured during the integrity verification process, so the computing platform needs to save these data.

In the system designed in this article, the SM3 algorithm is designed as an algorithm for the encrypted hash calculation. For a message m with a length of l (l < 264) bits, the SM3 hash algorithm is filled and iteratively compressed to generate a hash value. The length of the hash value is 256 bits:

(1) Filling: Assume that the length of the message m is l bits. First, add bit "1" to the end of the message, and then add k "0"s, where k is the smallest non-negative integer that satisfies l+1 + k ≡ 448mod512. Then add a 64-bit bit string, which is a binary representation of length l. The bit length of the padded message m' is a multiple of 512.

(2) Iteration: Group the padded message m′ according to 512 bits

$$m' = B^{(0)}B^{(1)} \ldots B^{(n-1)} \tag{1}$$

$$n = (l + k + 65)/512 \tag{2}$$

(3) Expand: Expand the message group $B^{(i)}$ to 132 words [W_0 , W_1,..., W_{67}],[w_0, w_1,..., w_{63}]:

- Divide the message group Bi into 16 words;
- FOR j = 16 To j = 67:

$$W_j \leftarrow P_1\left(W_{j-16} \oplus W_{j-9} \oplus \left(W_{j-3} \lll 15\right) \oplus \left(W_{j-13} \lll 7\right)\right) \oplus W_{j-6}. \tag{3}$$

$$w_j \leftarrow W_j \oplus W_{j+4}. \tag{4}$$

(4) Compression. Let A, B, C, D, E, F, G, and H be word registers. SS1, SS2, TT1, and TT2 are intermediate variables; then the compression function $V^{i+1} = CF(V^i, B^i)$ is calculated as follows:

$$ABCDEFGH \leftarrow V^i. \tag{5}$$

FOR j = 0 To 63:

$$SS_1 \leftarrow \left((A \lll 12) + E + \left(T_j \lll j\right)\right) \tag{6}$$

$$SS_2 \leftarrow SS_1 \oplus (A \lll 12) \tag{7}$$

$$TT_1 \leftarrow FF_j(A, B, C) + D + SS_2 + w_j \tag{8}$$

$$TT_2 \leftarrow GG_j(E, F, G) + H + SS_1 + W_j \tag{9}$$

$$D \leftarrow C \tag{10}$$

$$C \leftarrow B \lll 9 \tag{11}$$

$$B \leftarrow A \tag{12}$$

$$A \leftarrow TT_1 \tag{13}$$

$$H \leftarrow G \tag{14}$$

$$G \leftarrow F \lll 9 \tag{15}$$

$$F \leftarrow E \tag{16}$$

$$E \leftarrow P_0(TT_2) \tag{17}$$

END FOR

$$V^{i+1} \leftarrow ABCDEFGH \oplus V^i. \tag{18}$$

(5) Iteration: Iterate on m′ according to the following calculation steps:

FOR i = 0 to n-1,

$$V^{i+1} = CF\left(V^i, B^i\right). \tag{19}$$

END FOR

Among them, CF is the compression function, V^0 is the initial value of 256 bits, B^i is the packed group, and the compressed result of n iterations is V^n.

$$V^0 = 7380166f\ 4914b2b9\ 172442d7\ da8a0600\ a96f30bc\ 163138aa\ e38dee4d\ b0fb0e4e \tag{20}$$

Trust transfer follows the following idea: first measure, then verify, and finally jump. Starting from the root of trust, each currently running component first measures the next-level component to be run next, and then verifies its security according to the metric value. If its integrity meets the requirements, then the current-level component can jump to the lower-level component after running. Otherwise, it means that the lower-level components are unexpected, and the establishment of the trust chain is suspended.

4.3 Running Mode of the Embedded Trusted Computing

Based on the abovementioned static measurement technology, trust can be transferred from the root of trust to the uppermost application layer software to establish a complete static trust chain system. The construction of the static trust chain of the computer system is mainly divided into hardware startup code (BootROM or BIOS), Bootloader, and operating system. The idea of establishing the trust chain in each stage is as follows: after gaining control, measuring the code that will be run at the next level, and then extending it to the corresponding secure storage on the secure chip. The static chain of the trust system is generally divided into two stages: the first stage is trusted boot, which is responsible for safely booting the system to the operating system kernel, and the second stage is operating system integrity measurement, which is responsible for measuring the executables of the operating system kernel and application layer components.

4.3.1 Trusted Boot

Trusted boot uses a static metric mechanism to check the trustworthiness of each stage of the OS booter, configuration, OS kernel image, *etc*., and stores the metric results in the security chip in order, thus ensuring the security of the OS pre-boot environment. Taking Linux Grub as an example, the main steps of a general trusted boot are as follows:

(1) When the system is powered on, the hardware startup code (BootROM or BIOS) measures the first stage code of the

platform Bootloader, that is, the Grub Stage1 code, and the measurement results are saved to the integrity storage of the security chip, and then loaded into Stage1 and executed.
(2) Grub Stage1 loads and measures the first sector in Grub Stage1.5 and expands to the integrity storage of the security chip, and then executes Stage1.5.
(3) After Grub Stage1.5 gains control, it loads and measures Grub Stage2, and expands to the integrity storage of the security chip, and then transfers control to Grub Stage2. At this point, the Bootloader grub has been fully started and can perform operating system-related loading tasks.
(4) Grub measures its configuration file grub.conf and extends it to the integrity storage of the security chip, and then measures the operating system kernel that needs to be loaded, which finally verifies the overall integrity of the kernel file.

The trusted boot system ensures the security of the OS loader itself and prevents attackers from injecting the malicious code before the OS boots. All of these lay a secure foundation for the OS boot. The trust chain construction method of the trusted boot system is similar, and its main goal is to ensure the integrity of the Bootloader's own code, the Bootloader configuration file, and the OS kernel image.

4.3.2 Chain of Trust at the Operating System Layer

The operating system trust chain construction system is used to transfer the trust established by the trusted boot to the operating system and even the application program. To ensure the security of the trust chain construction, the measurement agent of the operating system trust chain is generally implemented in the OS kernel. The operating system kernel is executed from the decompressed image, and the OS measurement agent calls the TPM to measure the kernel modules, kernel service programs, *etc*., in turn when various modules of the OS are loaded according to the execution process of the operating system, thereby completing the construction of the operating system trust chain. After the operating system is started, in order to ensure the safety of running applications, the simplest idea is that each program can be executed after being measured by the OS measurement agent when it is loaded. This trust chain construction method can also cooperate with the program's black and white list mechanism to further enhance the security of system operation.

4.4 Logic of the Designed Trusted Computing Program

The core function of the trusted computing program is measurement and management. In the normal mode, the trusted computing program only measures executable programs. In the management and control mode, the trusted computing program simultaneously measures and controls executable programs. The overall program logic of these two modes is described in the following text.

4.4.1 Normal Mode Program Operation Logic

In the normal mode, when the actual power application is running on the power device, the program operation logic of the trusted computing program is shown in **Figure 4**, which is summarized as follows:

(1) The executable program of the power application is loaded and run by the system.
(2) The kernel module captures the loading behavior of the executable program by implanting HOOK functions.
(3) Before the specific operation, the captured executable program image is delivered to the trusted computing program measurement module.
(4) The trusted computing program measurement module invokes the function of the cryptographic module and uses the SM3 algorithm to measure the executable program image.
(5) The measurement results are added to the measurement log, and all the power applications that have been run by the system are recorded in the measurement log to form a snapshot of the current operating status of the power installation system.
(6) The kernel transmits the measurement log to the application layer measurement agent through the pipeline.
(7) The measurement agent further transmits the measurement log to the GUI interface through a communication mechanism (such as socket) for the administrator to observe the measurement log in real time and grasp the current operating status of the power device.

4.4.2 Operation Logic of Control Mode Program

In the management and control mode, the program operation logic of the trusted computing program is more complicated, as shown in **Figure 5**, which can be divided into two stages: the strategy generation stage and the specific operation stage.

4.4.2.1 Strategy Generation Stage

In the strategy generation stage (the lower part of **Figure 6**), the power device is initially installed and in an initial credible state. The administrator uses the knowledge base collection tool of the trusted computing program to formulate the whitelist. The program logic of the strategy generation stage is summarized as follows:

(a) The initial installation of the power device is completed, and the expected operating power application is deployed to the device according to the application scenario requirements of the device.
(b) The administrator enters the path where the expected power application is located and calls the trusted computing program knowledge base collection tool.
(c) The knowledge base collection tool will recursively search for executable programs in the specified path, call the SM3 measurement algorithm of the cryptographic module to hash all the executable programs, and add all the generated measurement results to the knowledge base to generate "expected executable programs knowledge base."
(d) The administrator operates the knowledge base by adding, deleting, modifying, checking, *etc*., and it sets the executable programs that are allowed to run on the power device to be

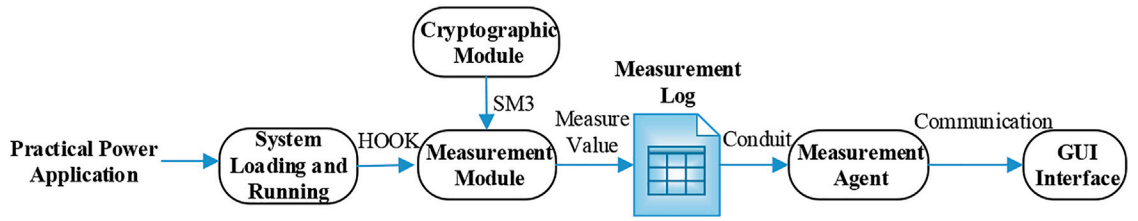

FIGURE 4 | Program running logic of the trusted computing program in the normal mode.

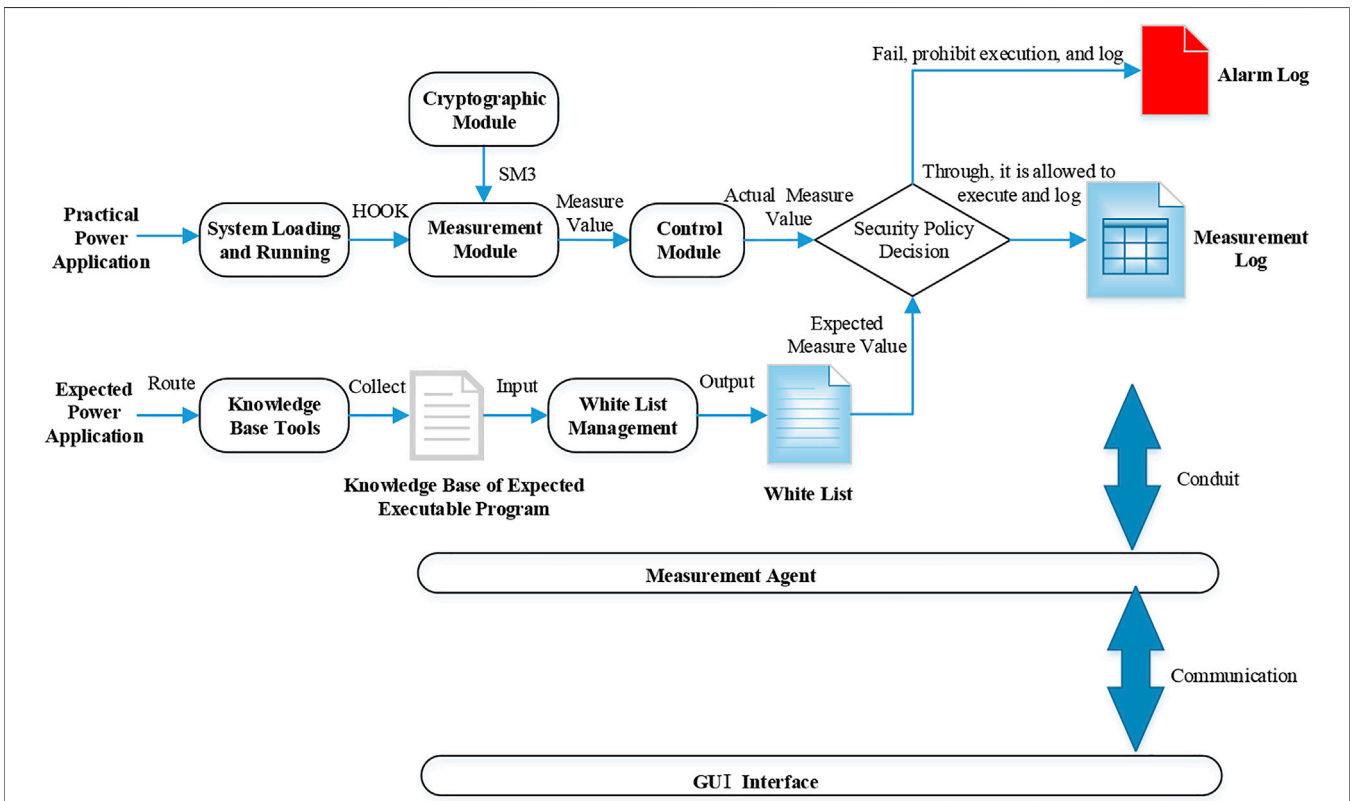

FIGURE 5 | Program operation logic of the trusted computing program in the control mode.

trusted, and the programs that are not allowed to run on the power device to be untrusted. The whitelist of the execution program is used by the kernel management and control module to execute the security policy.

4.4.2.2 Specific Operating Stage

In the specific operation stage (as shown in the upper part of **Figure 5**), the power device is already in the daily operation stage, and may be infected with viruses or be implanted with the malicious code. At this time, the trusted computing program will play a safe role in the control mode. The specific program logic at this stage is summarized as follows:

(a) The executable program of the actual power application is loaded and run by the system.

(b) The kernel module captures the loading behavior of the executable program by implanting HOOK hooks.

(c) Before the specific operation, the captured executable program image is handed over to the trusted computing program measurement module.

(d) The trusted computing program measurement module calls the function of the cryptographic module and uses the SM3 algorithm to measure the executable program image.

(e) The trusted computing program measurement module transmits the real-time measurement value to the trusted computing program management and control module.

(f) The trusted computing program management and control module enters the integrity verification program. The main logic is to match the actual measurement value of the measurement module with the expected measurement

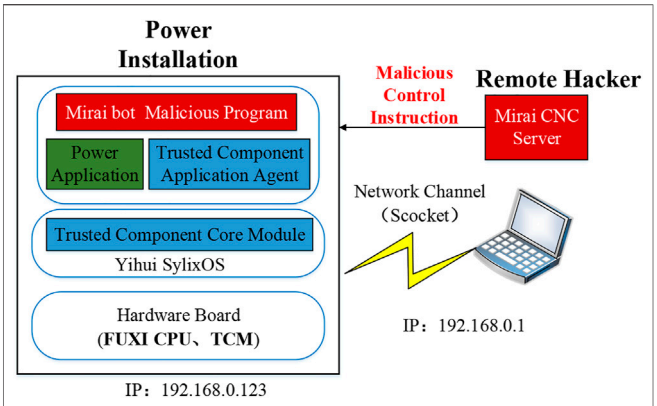

FIGURE 6 | Test environment.

value in the whitelist and give a security judgment based on the matching result. Those that meet the whitelist policy are considered acceptable and credible, and those that do not meet the whitelist strategy are considered untrustworthy.

(g) If it is judged to be credible, it means that the actual power application executable program has passed the security policy, and the management and control module will execute it to allow it to execute, restore its loading and operation to the normal system process, and record the measurement value in the measurement log.

(h) If it is judged to be untrustworthy, it means that the actual power application executable program has not passed the security policy. The control module will take over the normal system operation process, prohibit the program execution, and treat the attempted execution of the program as a malicious event, and the relevant information is recorded in the alarm log.

(i) Data generated by the trusted computing program such as whitelist, measurement log, and alarm log will be passed to the application layer measurement agent through the pipeline. The measurement agent further transmits the whitelist and log information to the GUI interface for display through other communication mechanisms (such as socket) for the administrator to view.

(j) At the same time, the GUI interface can receive the control instructions of the administrator and pass them to the measurement agent through the communication mechanism. The measurement agent is further passed to the trusted computing program of the kernel layer through the pipeline. The trusted computing program can complete the desired operation according to the control instruction, such as setting whitelist, set control mode, and clear blank list.

5 EXPERIMENTAL VERIFICATION

The experimental environment is shown in **Figure 6**. It consists of four parts: power terminal, remote hacker, management and control terminal, and debugging terminal. The four parts are interconnected through switch networking.

The power terminal used in the experiment is a distribution terminal unit (DTU) device, which is composed of hardware plug-ins, operating system, and application programs. It is developed based on the FUXI chip that is reached and developed by the China Southern Power Grid Digital Grid Research Institute, and the SylixOS operating system. The device is embedded with the built-in credibility component core module and trusted computing component application agent. In this experiment, Mirai CNC server simulates the remote attack hackers who access the network. The control terminal is used to enable control of the trusted computing components and display the measurement value of the measured program in the measurement log and alarm log.

The test environment is shown in the following text:

The trusted computing component is installed in the operating system kernel, as shown in **Figure 7**.

The authors have selected a dataset consisting of multiple executable programs. Some programs in this dataset are commonly used as power applications, some programs are third-party software libraries (such as OPENSSL), and some programs are simulated as malicious codes. For this dataset, a stress test script TCC_CompressionTest.sh is written, which selects programs from the dataset to trigger running every 10 s, and runs the script for a period of time to test the detection rate of illegal programs by the trusted computing program.

$$\text{Illegal program detection rate} = \frac{\text{the actual number of illegal programs detected n}}{\text{the test number of illegal programs injectedN}} \times 100\%$$

A sample set includes legal programs, illegal programs, and unknown programs, which is constructed on the actual power device. Through the brute force cycle test, the program is randomly selected from the sample set to trigger and run every 10 s, and it runs continuously, and then the effect of the trusted computing program on illegal programs is checked.

The components are tested by brute force injection of illegal programs. The test methods are as follows:

First, testing the real-time measurement function of the trusted computing component by running the env program that displays the operating environment of the system, which is shown on the QE interface. As we can see in **Figure 8**, the measurement value of the env program and the measurement value of the dependent library file libvpmpdm.so are all captured in real time.

Next, testing the security protection capabilities of the tested DTU device under malicious program operation. The implementation process is as follows:

Implanting a malicious program “portmap.cid” disguised as a configuration file through the MMS protocol to demonstrate the isolation function of the power terminal against malicious programs. The power terminal is interconnected with the monitoring system through the MMS protocol, and through the file service function provided by the MMS protocol, the

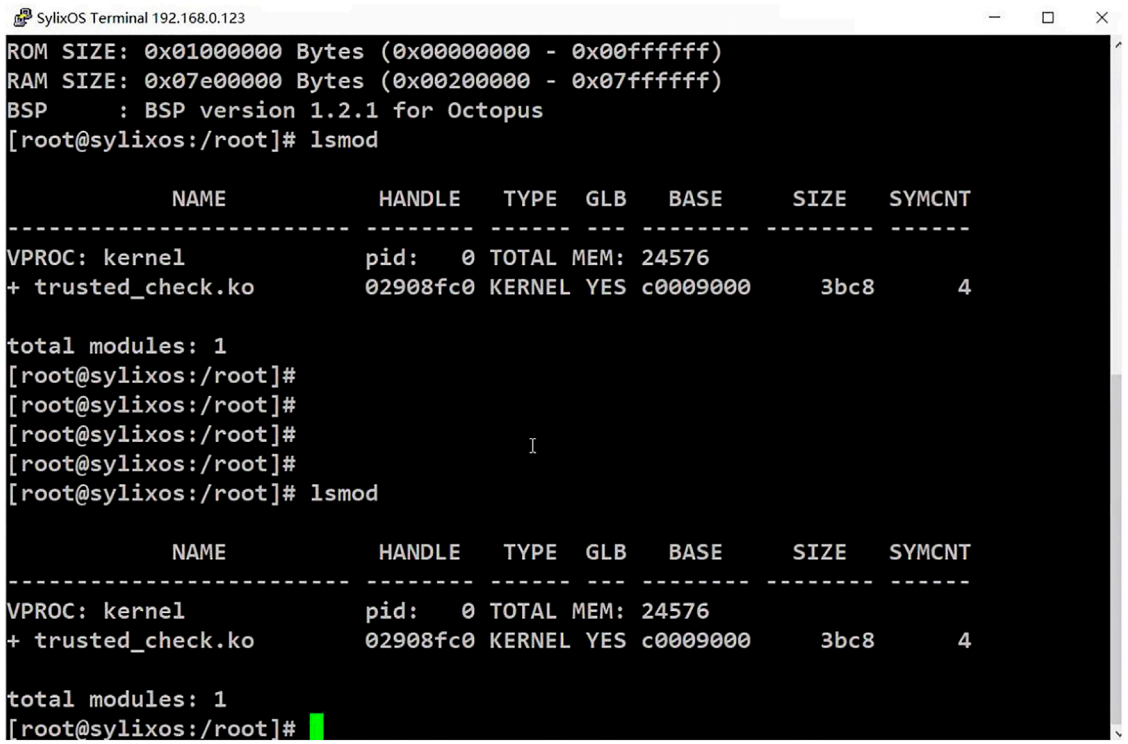

FIGURE 7 | Trusted_check.ko.

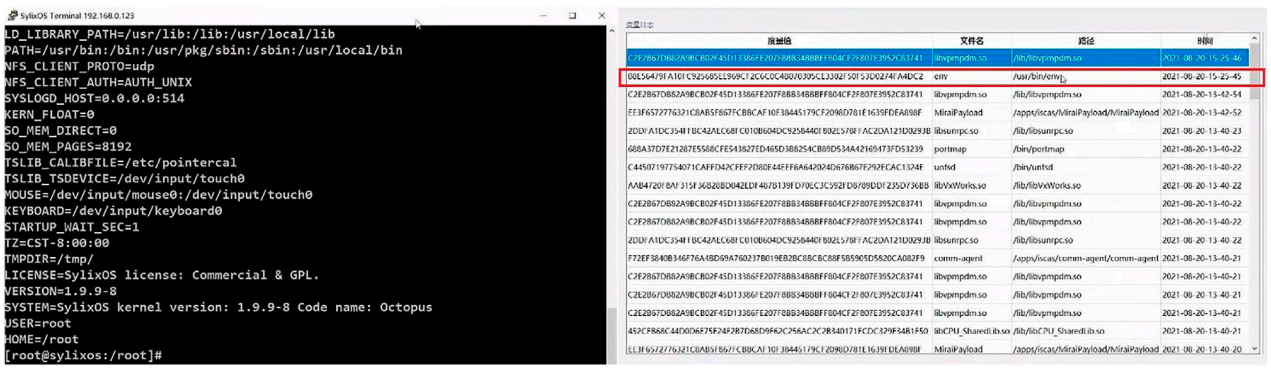

FIGURE 8 | Measurement value of the env program and the library file libvpmpdm.so.

malicious program disguised as a normal file "portmap.cid" is implanted in the device and called for operation.

It can be seen from **Figure 9** that "portmap.cid" is identified as an illegal program since there exists difference between the measurement value of "portmap.cid" disguised as a normal file and the measurement value of the real portmap.cid in the whitelist. The trusted computing component isolates the malicious program, and the malicious program cannot run.

In order to evaluate the impact of integrity measurement on the operating system loading and application loading of relay protection devices, the time performance of using the SM3 algorithm to obtain encrypted summary information is tested. In order to standardize test indicators, the integrity of 10KB data is measured uniformly, and the measurement time is recorded. **Figure 10** shows the results of 100 tests.

It can be seen from **Figure 10** that in 100 tests, the highest value is 0.719 ms/10KB and the lowest value is 0.586 ms/10KB, both less than 1 ms, indicating that the integrity measurement will not have a great impact on the system startup time and application loading time.

According to the test process described previously, an illegal virus implanted program disguised as a configuration file is transmitted to the DTU device through the MMS

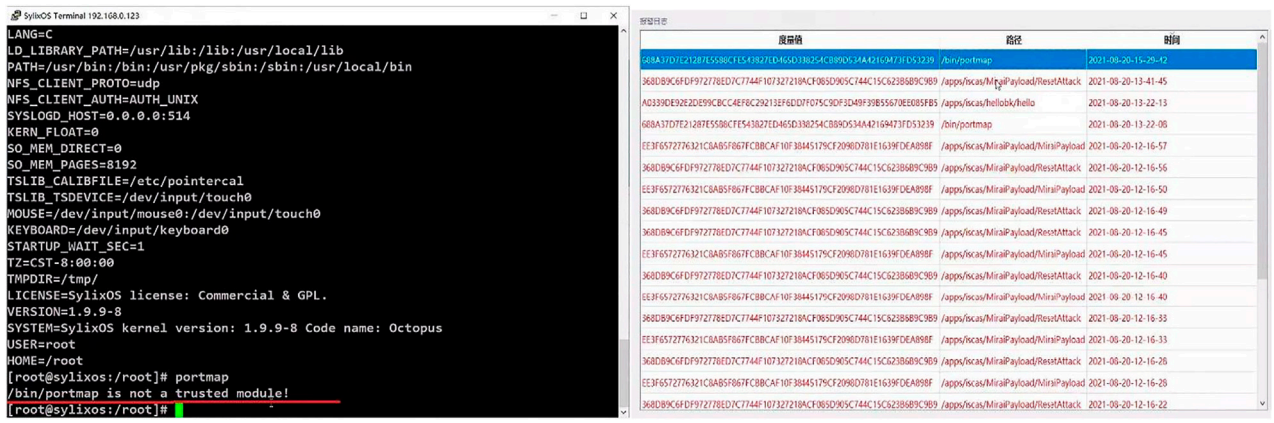

FIGURE 9 | A malicious program "portmap.cid" is identified as an illegal program.

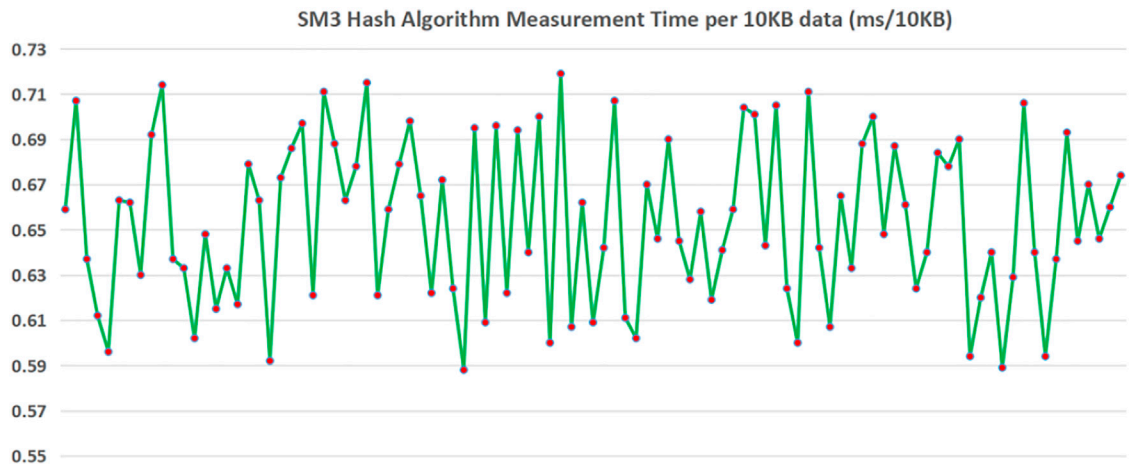

FIGURE 10 | A malicious program "portmap.cid" is identified as an illegal program.

protocol and loaded. In 300 repeated tests, the illegal program is detected 300 times, indicating that the detection rate of illegal programs by a test is 100%. Actually, on the premise that the physical security of the static root of trust is guaranteed, the trust chain is trusted to execute in the expected order of bootloader, OS, and APP, and all illegal programs can be detected.

6 CONCLUSION

With the advancement of the construction of the Energy Internet, a large number of terminal devices and multiple users have been connected to the power grid, gradually forming an open and interactive network environment, which has greatly changed the existing state grid's hierarchical and partitioned information security protection pattern with the vertical encryption and horizontal isolation. Once a threat invades, is adsorbed to the terminal equipment, or breakthrough the boundary protection, it will be unimpeded, and it is easy to cause major power grid safety accidents. In summary, this article proposes a network security protection scheme for power system embedded devices based on chip-level trusted computing. The main technical contributions are as follows:

(1) Designing the overall architecture of the chip-level trusted computing for power equipment.
(2) Proposing a method of using the on-chip SRAM of the main control chip as the PUF to construct the root of trust.
(3) Designing the program logic of the trusted computing component.
(4) Transplanting and applying the self-developed FUXI chip of the China Southern Power Grid and Sylix OS operating system in a distribution automation DTU device, and conduct the performance tests and safety protection tests.

The power system terminal chip-level trusted computing security protection solution developed in this article is suitable for the management needs of the future smart grid information security, and provides technology assurance for the construction of an open power business ecosystem with Chinese domestic chips as the core and the implementation of power Internet of Things technology.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material. Further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

WX contributed to the conception of the study, YY performed the experiment, QF and HY contributed significantly to analysis and manuscript preparation, WX and XL wrote the manuscript, and TC helped perform the analysis with constructive discussions.

FUNDING

This work was supported by the Science and Technology Project of China Southern Power Grid Digital Power Grid Research Institute (Grant No. 670000KK52200002).

REFERENCES

3rd Generation Partnership Project(3GPP) (2018b). *3GPP TS 33. 163 Battery Efficient Security for Very Low Throughput Machine Type Communication (MTC) Devices (BEST) V16. 0.0*. http://www.3gpp.org/ftp/Specs/archive/33-series/33.163/33163-g00.

3rd Generation Partnership Project(3GPP) (2018a). *3GPP TS 33. 220 Generic Authentication Architecture (GAA):generic Bootstrapping Architecture (GBA) V15.3.0*. http://www.3gpp.org/ftp/Specs/archive/33-series/33.220/33220-f30.

Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans.Dependable Secure Comput.* 1, 11–33. doi:10.1109/tdsc.2004.2

Bedi, G., Venayagamoorthy, G. K., Singh, R., Brooks, R. R., and Wang, K.-C. (2018). Review of Internet of Things (IoT) in Electric Power and Energy Systems. *IEEE Internet Things J.* 5 (2), 847–870. doi:10.1109/JIOT.2018.2802704

Ciavarella, S., Joo, J.-Y., and Silvestri, S. (2016). Managing Contingencies in Smart Grids via the Internet of Things. *IEEE Trans. Smart Grid* 7 (4), 2134–2141. doi:10.1109/TSG.2016.2529579

Efe, A., and Güngör, M. O. (2019). The Impact of Meltdown and Spectre Attacks. *Int. J. Multidiscip. Stud. Innovative Technol.* 01, 38–43.

Feng, D. G., Liu, J. B., Qin, Y., and Feng, W. (2020). Trusted Computing Theory and Technology in Innovation-Driven Development. *Sci. Sin.* 50 (08), 1127–1147. doi:10.1360/ssi-2020-0096

Feng, D. G., Qin, Y., Wang, D., et al. (2011). Research on Trusted Computing Technology. *J. Comput. Res. Dev. Chin.* 48 (8), 1332–1349.

Feng, W., Qin, Y., Feng, D. G., Yang, B., and Zhang, Y. J. (2015). Design and Implementation of Secure Windows Platform Based on TCM. *J. Commun.* 36 (8), 91–103.

Guo, Q. L., Xin, S. J., and Sun, H. B. (2016). Comprehensive Security Assessment for a Cyber Physical Energy System:a Lesson from Ukraine Blackout. *Automation Electr. Power Syst. Chin.* 40 (5), 145–147.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and Other Botnets. *Computer* 50 (7), 80–84. doi:10.1109/mc.2017.201

Kylanpaa, M., and Ekberg, J.-E. (2007). *Mobile Trusted Module (MTM)-an Introduction*. Ecoli http://research.nokia.com/files/NRCTR2007015.pdf.

Li, D. (2019). Analysis of Stuxnet Virus Incident and Enlightenment of Improving Industrial Control Security Protection Ability. *Netw. Secur. Technol. Appl.* 01, 9–10+24. (in Chinese).

Liu, R., Vellaithurai, C., Biswas, S. S., Gamage, T. T., and Srivastava, A. K. (2015). Analyzing the Cyber-Physical Impact of Cyber Events on the Power Grid. *IEEE Trans. Smart Grid* 6 (5), 2444–2453. doi:10.1109/tsg.2015.2432013

Liu, Z. L., Liu, B. J., Lu, Z. J., and Tong, Q. L. (2015). FPGA Design of Low Resource Consumed Arbiter PUF. *J. Huazhong Univ. Sci. Technol. Sci. Ed.* 44 (02), 5–8+14. (in Chinese).

Mispan, M. S., Halak, B., Chen, Z., et al. (2015). "TCO-PUF: a Subthreshold Physical Unclonable function[Conference Presentation]," in 2015 11th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME) (Glasgow, UK.

National Development and Reform Commission of China (2014). *Development and Reform Commission Order No. 14 Electric Power Monitoring System Safety Protection Regulations*. [Standard].

Office of State Commercial Cipher Administration (2006). *Block Cipher for WLAN Products-SMS4*. http://www. oscca.gov.cn/Up File/200602101642319790. pdf.

Office of State Commercial Cipher Administration (2012). *SM2 Elliptic Curve Public Key Cryptography Algorithm*. http://www. oscca.gov.cn/Up File/2010122214822692.pdf.

Office of State Commercial Cipher Administration (2010). *SM3 Hash Cipher Algorithm*. http://www.oscca.gov.cn/index.htm.

Park, L., Lee, C., Kim, J., Mohaisen, A., and Cho, S. (2019). Two-Stage IoT Device Scheduling with Dynamic Programming for Energy Internet Systems. *IEEE Internet Things J.* 6 (5), 8782–8791. doi:10.1109/JIOT.2019.2923432

Song, J. (2019). "Level-shifter Current Influence to Power Loss of Gate Driver IC [Conference Presentation]," in International Exhibition and Conference for Power Electronics, Intelligent Motion, Nuremberg, Germany (Renewable Energy and Energy Management).

Srinivas, J., Das, A. K., Li, X., Khan, M. K., and Jo, M. (2021). Designing Anonymous Signature-Based Authenticated Key Exchange Scheme for Internet of Things-Enabled Smart Grid Systems. *IEEE Trans. Ind. Inf.* 17 (7), 4425–4436. doi:10.1109/TII.2020.3011849

Sun, C., Liu, D., Ling, W. S., and Lu, Y. M. (2014). Research on Trustiness of Remote Terminal Units in Distribution Automation. *Power Syst. Technol.* 38 (3), 736–743.

Trusted Computing Group (2017b). *TCG Software stack(TSS) Specification. Version 1.10*. Ecoli http://www.trustedcomputinggroup org/developers/software_stack.

Trusted Computing Group (2017a). *TCG Specification Architecture Overview. Version 1. 2*. E.coil. https://www trustedcomputinggroup.org.

Trusted Computing Group (2017c). *TNC Architecture for Interoperability*. E coli http://www trusted-computing group org/resources/tnc_architecture_for_interoperability_specification.

Wang, Z. N. (2018). *Research on Information Security of Electric Energy Data Acquirsition Terminals Based on Trusted Computing*. Beijing, China: North China Electric power university.

Xu, R. H. (2014). *The Research and Implementation of Embedded Power Distribution Terminal Based on Trusted Computing*. Beijing, China: North China Electric power university.

Yu, H. Y., and Guan, C. L. (2021). The Cronier Security Incident in the U.S. A Warning to My Country's Energy Security Resilience. *Energy(in Chin.* 6, 26–29.

Zhang, L. (2019). Research on Trusted Computing Technology for Collection Terminal under Power IoT. *Electron. Components Inf. Technol.* 3 (12), 113–116.

Zhang, S. M., Wang, Z. N., and Wang, B. Y. (2017). Terminal Integrity Detection Scheme of Electricity Information Acquisition System Based on Trusted Computing. *Electr. Power Autom. Equip.* 37 (12), 60–66.

Zhang, T., Zhao, D. Y., Xue, F., et al. (2019). Research Framework of Information Security Protection Technology for Intelligent Terminals in Power System. *Automation Electr. Power Syst. Chin.* 43 (19), 1–8.

Zheng, T. F. (2019). *Research on Information Security of Smart Meter Based on Trusted Computing*. Beijing, China: North China Electric power university.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.