zPad

POWER-AWARE SECURITY DEVTEAM

Welcome to Banana Computer's zPad prototype development team!

The zPad is a highly-mobile tablet device, connecting users and critical systems at home, work, and in public. The zPad is similar to other tablet designs, but has key innovations:

Energy usage: There is no Moore's Law for batteries. Battery technology has not improved as rapidly as the computer technology supporting the zPad, and advances in battery technology are typically used to enable smaller devices, rather than larger power capacity. Since we can't rely on better (or larger) batteries to extend runtime, efficient use of power is one of the key design goals for the zPad team. The first decision made to support efficiency was the choice of platform: Intel's Atom. The Atom is Intel's newest line of efficiency-oriented processors, and competes with offerings from other manufacturers for the use in the netbook, tablet, mobile Internet device (MID), and embedded markets.

Security: Mobile and embedded devices often carry and process sensitive data and are used in insecure and highly dynamic environments, where attacks are becoming increasingly sophisticated. Accordingly, the second major goal for the zPad's software is an emphasis on security.

Like all features, security costs computational cycles; cycles cost power; and battery-powered devices have none to spare. While Banana Computer has always emphasized secure system software, consumers demand longer battery life as they grow more dependent on mobile devices. This creates a subtle tension between security and power use.

Banana Computer is interested in determining if a sufficient level of security can be maintained – while simultaneously reducing power consumption – by modulating system behavior based on power state and security posture. To investigate this issue, Banana Computer has assembled a crack team of security and software experts to develop and test components of a new Power-aware Security framework.

Power-aware Security

When power is "unlimited" -- such as when a device is plugged into a wall socket -- security engineers typically prefer to always apply security features “just to be on the safe side.” Additionally, due to their importance, security services are designed to provide security functionality without compromise – if efficiency is considered, it is typically as an afterthought and in regards to runtime. As a result, security features can consume a significant amount of power. In contrast, Power-aware Security involves carefully identifying the security and power-use properties of different security operations, and modulating behavior based on security requirements and the current power level. Modulating behavior can reduce power consumption, but can also reduce security. The importance and interconnectedness of security and power requires designers to carefully balance the needs of both.

As a simple example, larger encryption keys provide stronger security, but require more computation. In some power-limited circumstances, smaller keys (or no encryption) could be acceptable, such as when the device is in a trusted operational environment and the data is not sensitive. On the other hand, if the device is in a high-threat situation with little power, it may require disallowing certain actions because the they simply cannot be adequately secured with the available resources.

In general, Power-aware Security unifies three main concepts:

Security Posture

"Security posture" is a term that encapsulates the many security-relevant properties of a system into one abstract entity. Many factors are involved in determining a security posture, such as the trustworthiness of the system software and hardware, the security of the network being used, the presence of security sub-systems such as firewalls and antivirus, and so on. As an example, a system in an unsecured network has a worse security posture than one inside a protected corporate LAN.

Power State

The power state of a system describes the conditions relevant to energy consumption in the system. Battery power level, processor workload, signal strength and whether the device is plugged in are all power conditions.

Data Sensitivity

Data sensitivity describes the care that must be taken with different kinds of data. While we commonly think of confidentiality as the primary security property, integrity (ensuring that data cannot be inappropriately modified) and availability (ensuring that data is available when needed) are also important – sometimes more important. It is essential to understand the requirements resulting from the sensitivity of data when creating any kind of security policy – especially when the policy results in dynamic security behavior.

Three Themes of Power-aware Security

Based on these concepts, we have identified three main themes for the Power-aware Security initiative:

1. The Power-Security-Data Matrix

Design decisions relating to security and power can be placed in a plane with two dimensions: *power* and *security*. At the origin lies points which provide no additional security and consume no additional power. Moving away from the origin on the power axis, design points consume increasing amounts of energy. Moving away from the origin on the security axis provides increasing amounts of security. Design points can lie anywhere within this plane, with the quadrants representing "low security/low power," "low security/high power," "high security/low power," and "high security/high power."

Flexible, power-aware security policies and systems can be designed around this matrix with the addition of some conditions. For example, suppose a policy states that "high security" is always required for network

transmissions except when on a trusted network, and that "high security" requires "high power" (because of encryption's power requirements). The logical extension of these conditions is that transmissions in a "low power" state are only allowed when on a trusted network; that "high security/high power" is the default state, and that the system is not allowed to transmit data in a "high security/low power" condition.

Policies can grow more complex as the number or range of dimensions increase. Ranges can increase by adding additional levels, such as a third “medium” level to a system with only “high” and “low” options. The number of dimensions can increase by including other requirements, such as the sensitivity of the data in question. For example, if data sensitivity is added to the previous example, it could be that “low security” behavior is acceptable for “non-sensitive” data – regardless of the security posture of the system – but that transmitting “high-sensitivity” data is treated as always being on an untrusted network.

zOS's internal Security-Power-Data matrix represents the Security-Power-Data Matrix as follows: zOS identifies three threat levels to describe its environment: *Friendly*, *Unfriendly*, and *Hostile*. Similarly, it identifies three levels of sensitivity: *Unclassified*, *Classified*, and *Top secret*, and categorizes cryptographic strength with three levels: *Strong*, *Average*, and *Weak/None*. The zPad itself recognizes three battery levels: *High*, *Medium*, and *Low*. When the zPad is plugged in to a power source, it registers "Unlimited" power the maximum security is used.

By considering power, security, and data sensitivity, Power-aware Security systems can always choose the lowest-power behavior that still complies with the security policy for the relevant data.

2. Offloading Security

Offloading security is a specific approach for reducing security-related power cost. The main idea behind “offloading” is that if secure communication can be cheaper than computation, security computations may be able to be performed elsewhere more efficiently. In this approach, a battery-powered system enlists a remote server to perform computationally intensive, security-oriented tasks on its behalf. However, this depends greatly on the computations involved, the communication channels available, and the time-sensitivity of the related tasks. The question is, how much can be saved, and when?

3. Accurate Cost Analysis

Above all, the Power-aware Security initiative requires accurate cost analysis. Unfortunately, this is not as easy as it sounds.

Engineers and scientists cannot make informed decisions without accurate data. Previously, most researchers could only consider the runtime of tasks in order to estimate the amount of power consumed. However, empowered by the Atom-LEAP platform, engineers at Banana Computer can accurately measure the changes in power consumption of discrete components (such as CPU, hard disk, memory, and USB) resulting from any software execution, including those relating to security. However, accurately quantifying these effects requires careful experimental design and execution. Experiments must be designed to isolate interference from other system parameters, and must be repeated enough to provide statistically confident results.

Quantifying abstract costs is not always easy. While the Atom-LEAP allows us to identify when one algorithm "costs" twice as much power as another, it's much harder to quantify security. This is because the "cost" of security depends on the threats being quantified, the estimated loss, and the effect of the security choices made. For example, using a 4096-bit key instead of a 2048-bit key theoretically provides more security. However, how much more security it really provides depends on which will happen first: computers get fast enough to brute force the key, cryptanalysts discover a fatal flaw in the algorithm, or the universe ends.

Other costs are even "fuzzier," and can depend heavily on the user and other elements of the usage scenario. For example, how much is convenience or user frustration “worth?” If users are too frustrated by a policy, will they try to circumvent it -- or buy a competitor's product? Is it ever acceptable to send sensitive data unencrypted? Are industry standards for cryptography acceptable or is some “above and beyond” technique necessary for especially sensitive data? Other costs are practical; perhaps offloading cryptography to a fixed server is power efficient, but the turnaround time is much longer than if the computation was performed locally. Is the delay acceptable? Questions like these must be addressed for any design.

4. Power-aware Security Research Areas

Banana Computer executives want to explore Power-aware Security applications in five areas related to the zPad. You and your group members will select one of these tasks to investigate:

CryptoFlex -- Managing power/crypto/security automatically.

PowerSecZones -- Enabling applications to manage their own power consumption based on power level and threat.

OffLoading -- Reducing power use by moving security-related computation to a remote host.

ElectricSandbox – Measuring the cost of isolating systems with virtualization and other techniques.

CryptoDisk Evaluation – Measuring the cost of securing long-term storage?

CryptoFlex

1. Overview

Most systems that apply cryptography in order to protect data use a “one size fits all” strategy. In such an approach, all data is encrypted with the same algorithm, regardless of the power level of the device, the cost of the algorithm, the sensitivity of the data or the security properties of the environment. An unnecessarily costly defense is a poor use of resources.

CryptoFlex(tm) is a Power-aware Security feature scheduled for the zPad. CryptoFlex is a flexible, combined networking and cryptography stack that gives the operating system the ability to modify how cryptography is used to protect network transmissions. With CryptoFlex, power is saved by modulating cryptography based on a mandatory policy which balances the power state of the zPad with its security posture and the sensitivity of the data in question.

Groups investigating CryptoFlex will will be responsible for designing, testing, and evaluating *part* of CryptoFlex. This preliminary research will focus on identifying how CryptoFlex could work in a real system, by identifying cryptographic algorithms with differing power and security properties, designing a cryptographic change protocol, investigating scenarios where it could (or couldn't) save power while providing adequate security, and discussing the practical ramifications of CryptoFlex.

2. Specification

2.1 Investigation

Groups will need to identify the kinds of network traffic for which their solution will be suitable. For example, the zPad supports virtually every kind of network traffic imaginable. It can be used on a wired or wireless network, supports VOIP, virtual private networking, secure file exchange, instant messaging, and interactive shell sessions, and more. This includes traffic with both large and small packets, different sensitivities to latency, and so on. Some of these kinds of traffic may not be compatible with CryptoFlex – that's OK. Our job is to determine how, why, and when CryptoFlex works, and when it doesn't. We'll do that by investigating the costs of several cryptographic algorithms, designing a mechanism for changing cryptography on the fly, and building a prototype we can evaluate.

In order to do this, groups will need to identify cryptographic algorithms to support the different required levels of data security and power consumption. This will require testing the power consumed by specific algorithms when encrypting and decrypting data of various lengths.

Your group will also need to define a protocol to support the changes in cryptography in your CryptoFlex protocol. This protocol handles any communication between the sender and receiver necessary to support CryptoFlex. For example, at the bare minimum, the sender will need to be able to tell the receiver what cryptographic algorithm is in use for a particular transmission. If the design enables cryptographic change during transmission, some technique for changing the algorithm is necessary. While your prototype may not be fully functional, your protocol should be able to support all the proposed features in your CryptoFlex design.

Obviously, you must design this protocol carefully, and describe any known weaknesses in it. For example, can an observer identify a cipher change by viewing packets? Can a third party force a cipher change in any way? How is key exchange handled? Does the overhead of the cipher-changing system itself consume a meaningful amount of power?

2.2 Evaluation

Your prototype will be a simple file transfer application. This application will be given a comma-separated value (CSV) list of input describing file names, data sensitivities, security posture, and power level. Files will be sent one at a time. Your target network will determine how the packets are sent (i.e., many small packets, fewer

larger packets, TCP, UDP, etc.) Your tool should read each line, open each file, and based on the Security-Power-Data matrix information, send the file over the network to a receiver, who is able to decode the files correctly. The CSV file format is as follows:

File name	Sensitivity	Power	Security
File1.txt	Low	High:Medium	Friendly
File2.txt	Medium	Medium	Friendly:Hostile
File3.txt	High	Low	Hostile

Note: Transitions can occur in the middle of a file transfer. You may assume that two values for a single parameter represents a transition in the middle of the file. Three transitions happen at even thirds, and so on.

2.3 Analysis

In addition to analyzing the outcome of your investigation and prototype, your group should define a policy for CryptoFlex and analyze its effects. This includes more than a description of when to apply different cryptographic algorithms. For example, consider a scenario where a user is in a trusted environment, and is allowed to transmit sensitive data without encryption. However, the zPad is a mobile device; you will need to describe how the system should respond if the user leaves the secure environment. In another scenario, suppose a nurse in an ER has a document that is needed at a medical station in a legitimate emergency. However, CryptoFlex has determined that there may not be enough power to use the required level of encryption during transmission. What should happen? We suspect there are many unexpected scenarios like these. How should they affect the design, implementation, or application of CryptoFlex?

3. Deliverables

See the Deliverables section of this document for specific instructions.

PowerSecZone

1. Overview

It's important for the kernel to be able to modify it's behavior based on power level. However, user programs comprise a large portion of system workload. While the kernel can't arbitrarily change the behavior of most user programs, the programs themselves could be designed with awareness of their costs and with the ability to modify their behavior based on security posture, power level, and data sensitivity. That's what PowerSecZone is all about.

PowerSecZone is based on the Power-Security-Data matrix – the foundation of Power-aware Security. In order to make trade-offs based on power and security, the zPad needs detailed power level and security posture information. While zOS will not mandate specific behavior from user-level programs, the PowerSecZone facility will make power and security information available to user applications, enabling them to *voluntarily* modify their behavior so that they can reduce power consumption – extending battery life while maintaining an appropriate level of security.

Design teams working on PSZ will define interfaces for PowerSecZone information, and demonstrate how security-oriented applications could cooperate with the operating system to reduce power consumption.

2. Specification

2.1 Investigation

While this general model has been defined for PowerSecZone, an API for sharing this information with applications has not. The possibilities are endless; one option would be to export the data into resources userland programs could query, like the /proc filesystem in modern Unix systems. Your team will choose a design and evaluate an API for sharing PowerSecZone information. Another possibility would be to create a system call interface in the kernel; programs could query the state of the system by making a syscall and reading the results. API designs are likely to have both benefits and drawbacks.

Your team will also need to identify security applications that may be able to modulate their behavior to conserve power while still providing an acceptable level of security. While power reduction possibilities abound for normal applications (for example, a Word processor could disable or reduce the frequency of on-the-fly spell and grammar checking), security applications are unique in that they provide essential, time-sensitive functionality. It is not immediately clear which applications might be able to reduce their security-related power consumption without reducing the overall level of protection provided.

For these candidate applications, your team will identify specific strategies for security-oriented power modulation. This includes techniques to reduce power consumption, and identifying opportunities for applying those techniques. In terms of reducing power consumption using different kinds of cryptography is an obvious approach. Another approach is to defer security operations – or precompute them – whenever possible. (For example, battery life could be extended by delaying expensive security scans until the zPad is plugged in or until a maximum amount of time passes.) These strategies are not always possible due to situational constraints; what are these constraints?

Finding opportunities to apply these strategies is about leveraging the fact that the level of risk facing a system is nonuniform – some operations and environments are more or less secure than others. If a high-security environment is innately more secure, perhaps (for some set of applications) the same level of security can be achieved while spending less power on security operations. However, this is not always possible while maintaining security. Design teams working on PowerSecZone will explore this space by identifying and evaluating opportunities for using PowerSecZones.

2.4 Evaluation

Your group will need to design and run experiments to test your hypotheses and quantify the amount of power that could be saved by your techniques. In some cases, you may be able to use preexisting applications. In

other cases, you may have to implement “dummy” applications which are plausibly representative of your target applications. Then, you will use the Atom-LEAP platform to measure the power saved through the use of your techniques.

2.5 Analysis

Finally, your group will perform an analysis of your designs, in order to identify the kinds of security operations which are most and least costly, which types of operations can be safely deferred, and so on. Your analysis should also include recommendations for future research, testing, and deployment.

3. Deliverables

See the Deliverables section of this document for specific instructions.

ElectricSandbox

1. Overview

Process isolation is one of the most important services that an operating system provides to user applications. Isolation allows every process to act as though it had complete control of the computer, while simultaneously ensuring that processes cannot interfere with one another. However, process isolation is not perfect; it is difficult to provide truly comprehensive process isolation. As a simple example, processes can typically detect one another. More significantly, misconfigured or vulnerable applications regularly demonstrate the limits of process isolation by taking down an entire server and everything running on it.

“Sandboxing” refers to taking intentional steps to isolate processes running on a shared system. While preemptive multitasking and virtual memory can be thought of as types of sandboxing, the term typically refers to going “above and beyond” standard techniques. Virtualization has become a popular method for providing stronger isolation. Software virtual machines such as VMware, Virtualbox, User-Mode Linux (UML), Xen and others are popular methods for providing stronger isolation. In this approach, processes which formerly would have been run on the same logical and physical system are run instead in individual virtual machines on a single physical computer. In this way, if the process is corrupted, it can typically damage only its own virtual environment.

Other kernel-level facilities provide stronger isolation without running complete virtual machines. For example, “chroot jails” attempt to lock processes into a subset of the file system. This technique, while popular, is not foolproof. Similarly, many applications (such as Google's Chrome web browser) are moving away from a “single process, multiple thread” model to a multiple process model. This is because sibling threads share the address space of their parent process, which enables a thread to corrupt an entire process. Separate processes cannot do this unless there is a flaw in the OS.

Groups working on the ElectricSandbox project will thoroughly investigate the power and security trade-offs of various forms of sandboxing. Of course, sandboxing and other kinds of virtualization require more computation – in some cases, considerably more. As a result, it's reasonable to believe that virtualization has a heavy power overhead when compared to simpler sandboxing approaches (or no sandboxing at all). Additionally, while generic virtualization software can run on any system, special hardware and software support for virtualization exists. These attempt to reduce some of the runtime overhead of virtualization through special instructions and kernel interfaces. By reducing runtime, they should also reduce power consumption. However, it's not always that simple – these advanced support systems may draw more power in the process. Unfortunately, we don't know how much virtualization costs, and as a result, it's difficult to make informed decisions.

2. Specification

2.1 Investigation

The first step of this project is to investigate the basic power and security trade-offs of several different types of virtualization solutions. Groups should investigate at least the following: Virtualbox (which is based on QEMU and functionally similar to VMware), User-Mode Linux, multithreading versus multiprocessing, and chroot jails. For Virtualbox, groups should investigate the use of special hardware instructions if available. If they are not available, groups should attempt to quantify the expected performance loss or gain by the use of those instructions. Groups should compare and contrast the losses and gains resulting from these techniques.et

2.2 Evaluation

Once the first step is complete, your group will need to compare various techniques with a diverse set of workloads. To start, groups should test multithreading versus multiprocessing using a workload (creating it if necessary) that performs an identical amount of work but uses both techniques. For more abstract virtual machines, workloads should include the multithreading and multiprocessing workloads, but also more realistic applications and workloads. For example, you should include I/O heavy tasks such as database (MySQL) or filesystem (Modified Andrew Benchmark) benchmarks, computationally heavy tasks such as numerical

computing or CG-rendering, and networking-heavy tasks (to test the costs of virtualized networking). These tests should be performed with and without options that may change the power profile (such as hardware instructions). How do the power profiles compare across the board?

2.3 Analysis

In this portion of the project, your group should analyze, describe, and organize the different sandboxing techniques in terms of their costs and benefits. This should include a description of the kinds of tasks best suited for different virtualization techniques and other practical recommendations. Finally, your analysis should include a discussion of how the power consumption of the individual sandboxing techniques might be reduced. For example, if disk power consumption increases dramatically under Virtualbox, how might we be able to reduce consumption while still using that solution?

3. Deliverables

See the Deliverables section of this document for specific instructions.

OffLoading

1. Overview

“Passing the buck” is slang for avoiding something, with the expectation that someone else will pay the cost of dealing with it. Passing the buck often carries a negative connotation in society. On the other hand, the word “delegation” has many of the same meanings, but with fewer negative overtones. In fact, delegation is an essential part of most organizations. While many of the executives at Banana Computer were once engineers, their time is now better spent running the company. Instead of designing, they delegate design tasks to teams like yours. Similarly, delegation can reduce the power consumption on a system where energy is limited and thus more valuable (e.g., a zPad) by moving computation to a system where it is comparatively unlimited (e.g., a data center). This can prolong battery life so long as the expense of delegation is cheaper than the original task.

Where CryptoFlex explores modulating security-related power consumption based on system state, and PowerSecZone attempts to save power by (among other things) deferring security operations, OffLoading is an investigation of how and when security operations can be delegated across the network in order to prolong battery life. For example, suppose a zPad is accessing an SSL-encrypted website using a wireless access point that encrypts all wireless traffic automatically. In this scenario, the zPad is paying twice for encryption – once for the wireless, and once for SSL. Instead, how much power would the zPad save if the wireless access point served as a proxy? In this scenario, the access point, which has “unlimited” power, would maintain the end-to-end SSL connection, and the zPad would simply send and receive through the access point. What are the security risks in such a solution? We expect that there are many ways in which security-related computation can be delegated, but that for many of those opportunities the cost of delegation outweighs the power savings.

Groups working on the OffLoading project will identify and investigate a series of possibilities for delegation of security-related computation, and will evaluate the actual savings by running experiments using the Atom-LEAP.

2. Specification

2.1 Investigation

Some kinds of security-oriented computations can be offloaded to other systems, and some can't. In this portion of your project, you will identify and describe the kinds of operations which fall into both categories. In the course of this investigation, you should select a few applications or operations that you believe could be made more efficient by using an OffLoading approach. Additionally, you should perform any preliminary testing necessary before building your test applications.

2.2 Evaluation

In this portion of the project, you will build and test your selected applications and OffLoading strategies, in order to measure the power saved through the use of your techniques. In some cases, you may be able to reuse existing software, while in other cases you may need to create dummy applications with similar properties. It is essential that your technique maintains the security properties of the system. For example, the wireless-SSL proxy is only conceivable because the wireless connection already provides encryption for the data. Moving encryption to the access point, but not using encryption on the wireless traffic, is not acceptable without at least recognizing the risks inherent in that design. (Other issues include whether the wireless encryption is strong and whether it protects wireless users from each other, or only from outsiders.)

The result of this step should be quantitative data showing the potential power savings of your techniques, or alternatively, a careful explanation (based on the data) of why the strategy does not provide a reduction in power consumption.

2.3 Analysis

Finally, your group should provide a thorough analysis of your results, including the systems you tested as well as potential systems you identified in the Investigation portion of the project. After having done your

experimentation, how do you feel about the other potential opportunities? Are they more or less likely to succeed? What does your experience with this project tell you about the potential for power savings using OffLoading?

3. Deliverables

See the Deliverables section of this document for specific instructions.

CryptoDisk

1. Overview

The value of cryptography has been made very clear to society over the last few years, as repeated revelations of stolen sensitive data continue to make the news. One result of this newfound enthusiasm for cryptography has been a multiplicity of methods for encrypting persistent storage such as hardware and software-based full disk encryption (FDE). Banana Computer feels that it is necessary to include some encrypted storage options in the new zPad. However, they want to base the design choice on the security provided and the additional power consumption involved.

The goal of this project is to evaluate the various costs associated with supporting encrypted file storage, and to use this information to identify the best choice for the zPad. At the highest level, some analysis is required to measure the power cost of encryption and decryption, which is used for encrypted containers and software-based FDE. At the next level are issues regarding filesystem type and behavior. Buffering and journaling behavior can have significant additional power costs, even if they do not require many more disk operations. At another level includes questions regarding block and file size, which may have an impact on the efficiency of encryption and decryption.

Finally, there are issues at the physical level. Flash disks and mechanical disks have different power profiles and operational requirements, which may make more comparatively more power available for cryptography. Using networked storage over an encrypted channel could save power by reducing disk power consumption – but may result in unintended consequences such as unacceptable latency. Some devices (both flash and mechanical) include onboard hardware encryption. With these devices, encryption takes place in the device, not on the CPU. How does hardware encryption change the power cost of file encryption?

Groups working on this project will investigate some or all of these issues, perform an evaluation of real systems using the Atom-LEAP, and make a recommendation for how the zPad should support encrypted storage.

2. Specification

2.1 Investigation

As discussed in the overview, many factors affect the cost of supporting encrypted file storage. In the first part of this project, your group should identify and describe as many potential parameters as possible. Because there will probably be too many parameters to exhaustively test, your group should select a subset of these parameters for quantification using the Atom-LEAP.

2.2 Evaluation

In this part of the project, your group will execute a series of experiments designed to quantify the performance effects of the parameters you have selected. It is most important to quantify the power costs of the different solutions, but it is also important to consider the performance in terms of time and latency. Your evaluation should include the design and execution of various disk workloads within carefully selected environments.

2.3 Analysis

In the final stage of this project, your group will analyze the results of your investigation and experimentation in order to make a recommendation as to how the zPad should support encrypted file systems. The different options are likely to have different performance profiles or other characteristics, and your recommendations should acknowledge those issues. For example, it could be that networked file storage is a more efficient way to provide secure storage, however it comes with a potential cost in terms of latency and availability.

3. Deliverables

See the Deliverables section of this document for specific instructions.

Deliverables

See website for deadlines.

1. Project Design 20%

Once your group is assigned, your team will collaborate on a design document outlining your approach to the project. This document should include:

...a potential design meeting the specification in the project description. This does not need to be a comprehensive design, but it must provide some level of end-to-end functionality. You should indicate which parts you believe are straightforward, which parts you will investigate thoroughly, and which questions you plan to leave for "future work." You should discuss the expected ramifications of your design in terms of function, ease of use, security and efficiency.

... a plausible description of how your prototype will be implemented. We don't expect you to write enterprise-ready software. However, having an end-to-end functional prototype (as opposed to an abstract model) of at least part of your design is necessary as a proof by construction of the performance properties of your approach. We do not expect your prototype to be completely written from scratch – you are welcome to use open source tools (such as system utilities and cryptographic libraries) provided you properly attribute your work and obey any relevant software licenses. Your implementation plan should convince your instructor that you have identified all the necessary components, understand how they will work together, and possess the ability to assemble them. You should also include a project schedule describing how you intend to distribute your work over the quarter.

...an evaluation plan for your system. You need to describe how your team will evaluate the correctness, performance, and security of your system. Your proposed correctness evaluation will need to show that your design is sound, and that your prototype works as designed.

The security evaluation is intended to show first, that your design is not flawed, and second, to evaluate the security ramifications of your design (i.e., how security is lost or gained during execution).

Your performance evaluation will need to show the power costs of your system, both at a macro-level (which implies a choice of realistic workloads) and at a micro-level (by testing discrete components). You must use proper experimental technique, including thorough identification of system parameters and experimental factors, fair sampling, and calculating statistical confidence to a reasonable level. You must keep all data, clearly labeled according to how it was gathered, etc. This allows other researchers to re-evaluate and analyze your results.

Finally, it is critically important that your prototype does not *underestimate* the power cost of your design, because this falsely makes the performance of your system appear better than it should be. For example, neglecting to use encryption for all sensitive communication in your prototype (a security and correctness flaw) destroys the validity of your performance evaluation, because a correct implementation would undoubtedly consume more power.

Submit the design document for revision and approval before performing any other work.

2. Weekly Updates (40%)

Groups will meet weekly with the professor and TA to discuss progress. The whole group is required to come to each meeting, which will last approximately one hour.

3. Presentation 10%

Your team will give a group presentation to the rest of the zPad design team at the end of the quarter. The presentation should include each member of your group and must describe your investigation, evaluation, and analysis. Your presentation may include a demonstration if time permits. The length of the presentations will depend on the number of groups.

4. Paper 30%

You and your team will write a 10-15 page paper describing your project in detail and submit it before the end of finals week. Your report should include all of the above information, as well as a description of the development process (e.g., what worked and what didn't, design changes that were necessary, unexpected outcomes, etc.). Your report should include diagrams when appropriate, and must include appropriate graphical presentation of your evaluation data, such as histograms, line graphs, tables, etc. Your paper must provide proper attribution to sources.

Grading on all materials will be based on the quality of the research – a well supported, informative, *negative result* (such as showing that efficiency *cannot* be improved using a particular technique) can be just as important as an exciting positive result.